Spring 2020

Representations and representation learning for image aesthetics prediction and image enhancement

Michal Kucer
mxk7721@rit.edu

Representations and representation learning for image aesthetics prediction and image enhancement

by

Michal Kucer

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Chester F. Carlson Center for Imaging Science
College of Science
Rochester Institute of Technology

Spring 2020

Signature of the Author _____

Accepted by _____
Coordinator, Ph.D. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE

COLLEGE OF SCIENCE

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

Ph.D. DEGREE DISSERTATION

The Ph.D. Degree Dissertation of Michal Kucer
has been examined and approved by the
dissertation committee as satisfactory for the
dissertation required for the
Ph.D. degree in Imaging Science

_____

Dr. David Messinger, Dissertation Advisor

_____

Dr. David Ross, External Chair

_____

Dr. Christopher Kanan

_____

Dr. Alexander C. Loui

_____

Date

Representations and representation learning for image aesthetics prediction and image enhancement

by

Michal Kucer

Submitted to the
Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Doctor of Philosophy Degree
at the Rochester Institute of Technology

Abstract

With the continual improvement in cell phone cameras and improvements in the connectivity of mobile devices, we have seen an exponential increase in the images that are captured, stored and shared on social media. For example, as of July 1^{st} 2017 Instagram had over 715 million registered users which had posted just shy of 35 billion images. This represented approximately seven and nine-fold increase in the number of users and photos present on Instagram since 2012. Whether the images are stored on personal computers or reside on social networks (e.g. Instagram, Flickr), the sheer number of images calls for methods to determine various image properties, such as object presence or appeal, for the purpose of automatic image management and curation. One of the central problems in consumer photography centers around determining the aesthetic appeal of an image and motivates us to explore questions related to understanding aesthetic preferences, image enhancement and the possibility of using such models on devices with constrained resources.

In this dissertation, we present our work on exploring representations and representation learning approaches for aesthetic inference, composition ranking and its application to image enhancement. Firstly, we discuss early representations that mainly consisted of expert features, and their possibility to enhance Convolutional Neural Networks (CNN). Secondly, we discuss the ability of resource-constrained CNNs, and the different architecture choices (inputs size and layer depth) in solving various aesthetic inference tasks: binary classification, regression, and image cropping. We show that if trained for solving fine-grained aesthetics inference, such models can rival the cropping performance of other aesthetics-based croppers, however they fall short in comparison to models trained for composition ranking. Lastly, we discuss our work on exploring and identifying the design choices in training composition ranking functions, with the goal of using them for image composition enhancement.

Acknowledgements

I owe enormous thanks to my advisor Dr. David Messinger for his continuous support, patience, motivation, enthusiasm, and vast knowledge. At any time during my research I was excited about "a million" different ideas and projects at the risk of being hopelessly distracted. Dr. Messinger did a fantastic job finding balance between encouraging the pursuit of my interests and guiding me towards the completion of my PhD. I could not have imagined having a better advisor and mentor for my doctorate.

Tremendous thank you goes to my thesis committee: Dr. Chrisopher Kanan, Dr. Alexander Loui, Dr. David Ross for their insightful comments and discussions.

I am thankful to my friends: Baabak, Sagar, Will, Martin, Michal, Tereza, Kristan, Clay, and Laura, who've been there with me through thick and thin.

During my PhD, I was fortunate to conduct two incredible internships at the Los Alamos National Laboratory (LANL), and Naver Labs Europe. I would like to thank my mentors from LANL, Amanda Ziemann and James Theiler, and Naver, Naila Murray. Furthermore I would like to thank all of the incredible friends that I have made and still keep in touch with.

I have to thank the many friends and people at the Center for Imaging Science. First a big thank you to the fellow PhD students and office mates - Aneesh, Lauren, Mandy, Lucy, Sanghui, Jacob, Ryan just to name a few. Big thanks also goes to Marci, Susan, Beth, Joyce, and all of the staff in CIS.

Thank you to my family in the US - Sandy, Steve, Jeanne, Santosh, Jivan, Nisha, Shaun, Angad - and my host families from Indiana - the Sheeks and the Buchanans.

I would like to thank my parents Peter and Zuzana, who from young age served as my role models and stressed the importance of education. I want to thank my sister Zuzka for her constant encouragement. Without their support, I would not be where I am today.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

With continuous miniaturization of silicon technology and proliferation of consumer and cell-phone cameras, we have seen an exponential increase in the number of images that are captured [34]. Whether the images are stored on personal computers or reside on social networks (e.g. Instagram, Flickr), the sheer number of images calls for methods to determine various image properties, such as object presence or appeal, for the purpose of automatic image management and curation. One of the central problems in consumer photography centers around determining the aesthetic appeal of an image.

Aesthetics. Aesthetics is generally understood as the ”study of beauty”, though it is often challenging to pinpoint what beauty really is. Very often, beauty is viscerally experienced by a person and combines a variety of stimuli, emotions, etc.

Challenges. As the perception of aesthetics is a combinations of various stimuli, we run into the first challenge of aesthetic inference - subjectivity. Despite this hurdle, we know from previous work one can model an imprecise notion of objective beauty by combining opinions and preferences of several people, e.g. trying to predict the mean aesthetic score of the image. Such thing can be done well enough, and further utilized in other avenues, - e.g. in image cropping. Though this points to a possible avenue of modeling individual aesthetic preferences, which is challenging from the point of view of modeling and data-collection.

Potential uses of aesthetic inference. The ability to predict image aesthetics is important for several reasons: aesthetic scores can be used for (a) ranking images as a proxy for image quality, (b) enhancing image search and image retrieval, (c) education, (d) image enhancement, and (e) predicting other high-level image attributes (e.g. popularity, memorability, importance).

1.2 Contribution and outline

In this dissertation, we explore expert representations and representations learning approaches used in image aesthetics prediction and image enhancement. More specifically, we can summarize the main contributions of our work as following:

- We show learning architectures can be used to rank images according to aesthetics
  - expert features can aid deep learning representations
  - learning architectures can be adequately trained to rank image aesthetics
- we show aesthetic ranking functions can serve as an imperfect proxy for composition
- we show a dedicated composition ranking function is ideal for image cropping
- we outline generalized learning approach and good practices for training composition ranking functions.

The rest of the dissertation is organized as follows:

In **Chapter 2**, first we briefly discuss a typical computer vision pipeline, and then discuss the prior work on representations for aesthetic predictions. Lastly, we provide a brief overview of neural networks, and representation learning through pair-wise ranking optimization.

In **Chapter 3**, we present a method for multi-subject salient foreground detection, which can serve as a part of a pipeline for computing hand-crafted features. The content of this chapter is based on the algorithm presented in Kucer et al. [56].

In **Chapter 4**, we present our work whose goal is to bridge traditional approaches based on expert hand-crafted features and deep learning. The work focuses on understanding and evaluation of expert feature sets, and discusses their potential in improving convolutional neural network (CNN) features. The content of this chapter is based on the work presented in Kucer, Loui, and Messinger [57],

Chapter 5 presents an analysis of trade-offs in varying image size and network depth and their effects on the aesthetic ranking performance in resource-constrained CNN models. Additionally, we show that networks trained with pair-wise ranking methods can achieve near state of the art in aesthetic image cropping, though fall short as compared to models which aim to tackle related, yet different problem of composition ranking. The content of this chapter is based on the work presented in Kucer and Messinger [58].

Chapter 6 discusses our work on establishing good practices for learning composition ranking functions, in which we consider various aspects of the learning pipeline (data pre-processing, data sampling, architecture details, and loss functions).

In **Chapter 7**, we discuss conclusions from presented work and outline several open problems that deserve further attention.

Chapter 2

Background

In this chapter, we describe the relevant background related to image aesthetic inference. We discuss a standard pipeline in computer vision for extracting image features and using them to learn a function to predict a desired quantity. In the subsequent section, we present a summary of the previous work done on aesthetic inference. Lastly, we briefly discuss neural networks which currently dominate the approaches in several areas of computer vision, e.g. object detection and classification, face detection or speech recognition.

2.1 Standard Approach

The goal in image aesthetic assessment is to learn a function $f : X \rightarrow Y$, which given some image description $x = \{x_1, \ldots, x_i, \ldots, x_n\}$ (where $x_i = g_i(I)$ represents the i^{th} image feature), maps x to y. The output y can either be a label (e.g. a binary label denoting the image to be of high quality or low quality), multiple labels (e.g. a vector of features indicating image attributes like the rule of thirds), or a continuous score (e.g. a population average of a distribution of ratings). A standard approach for understanding aesthetic appeal (as well as other computer vision problems), can be seen in Figure 2.1, can be described as being composed of two parts: the feature extraction and decision phase. Feature extraction, is a step where one extracts a set of image descriptors $\{x_i = g_i(I)\}_i$. We can divide feature descriptors into following categories based on the function g_i: hand-crafted aesthetic features (features created with the aim of approximating various rules seen in images with high appeal and quality), generic (general features used in other computer vision tasks, e.g. Histogram of Oriented Gradients (HOG)), and learned (i.e. features learned as part of optimization of a neural network). In the decision phase, one uses the image descriptor, $\{x_1, \ldots, x_i, \ldots, x_n\}$, and feeds it into a previously learned machine learning model (e.g. Linear Regression, Boosted Decision Trees, or Convolutional Neural Network) to get the desired output.

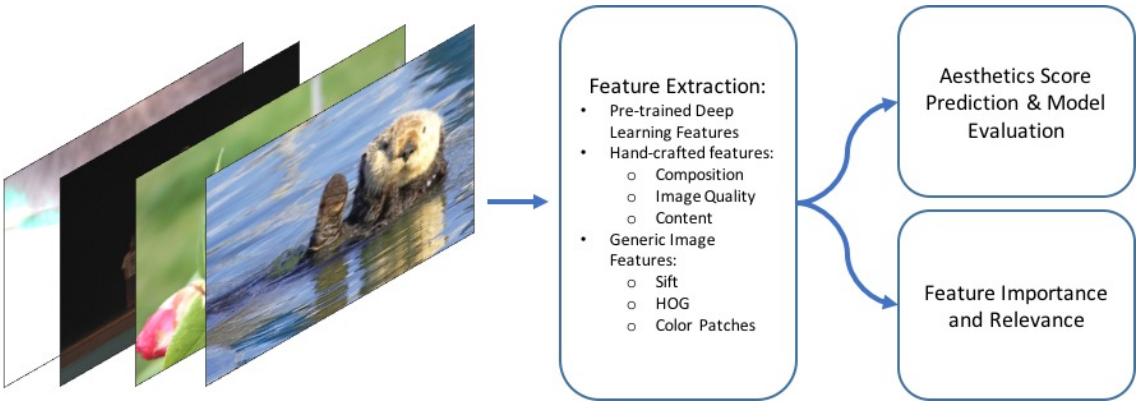


Figure 2.1: A standard pipeline used in image aesthetics problems.

2.2 Aesthetics Inference

In the recent years there is a fine line between studying photo quality and image aesthetics. Studies such as [47] use images rated on their aesthetics (in this case DPChallenge.com) to define their notion of quality by taking the top and bottom 10% of the images. Traditional notion of quality mostly includes perceptual qualities such as blur and color contrast. By choosing to define the quality of photos based on their aesthetics ratings, they inherently consider higher level semantic information otherwise relevant only to consideration of aesthetics. Therefore, the review will consider relevant papers that consider both the study of Image Aesthetics and Photo Quality as the main subject. In the following sections, we review possible sources of data for studying aesthetics and subsequently described previous work on aesthetics inference.

2.2.1 Datasets

In order to study the problem of image aesthetics, one has to address the feature representation that will represent the images and the learning paradigms one will employ to infer the value of unseen images. A separate issue is acquiring appropriate datasets, since aesthetics inference is a supervised learning problem requiring aesthetic labels. The datasets can come from either of two resources: controlled studies and community-contributed resources (e.g. media-sharing networks). Below we briefly describe the main sources where one can obtain labeled images that can be used for aesthetics inference. In Figure 2.2, we show examples of images used for aesthetics inference.

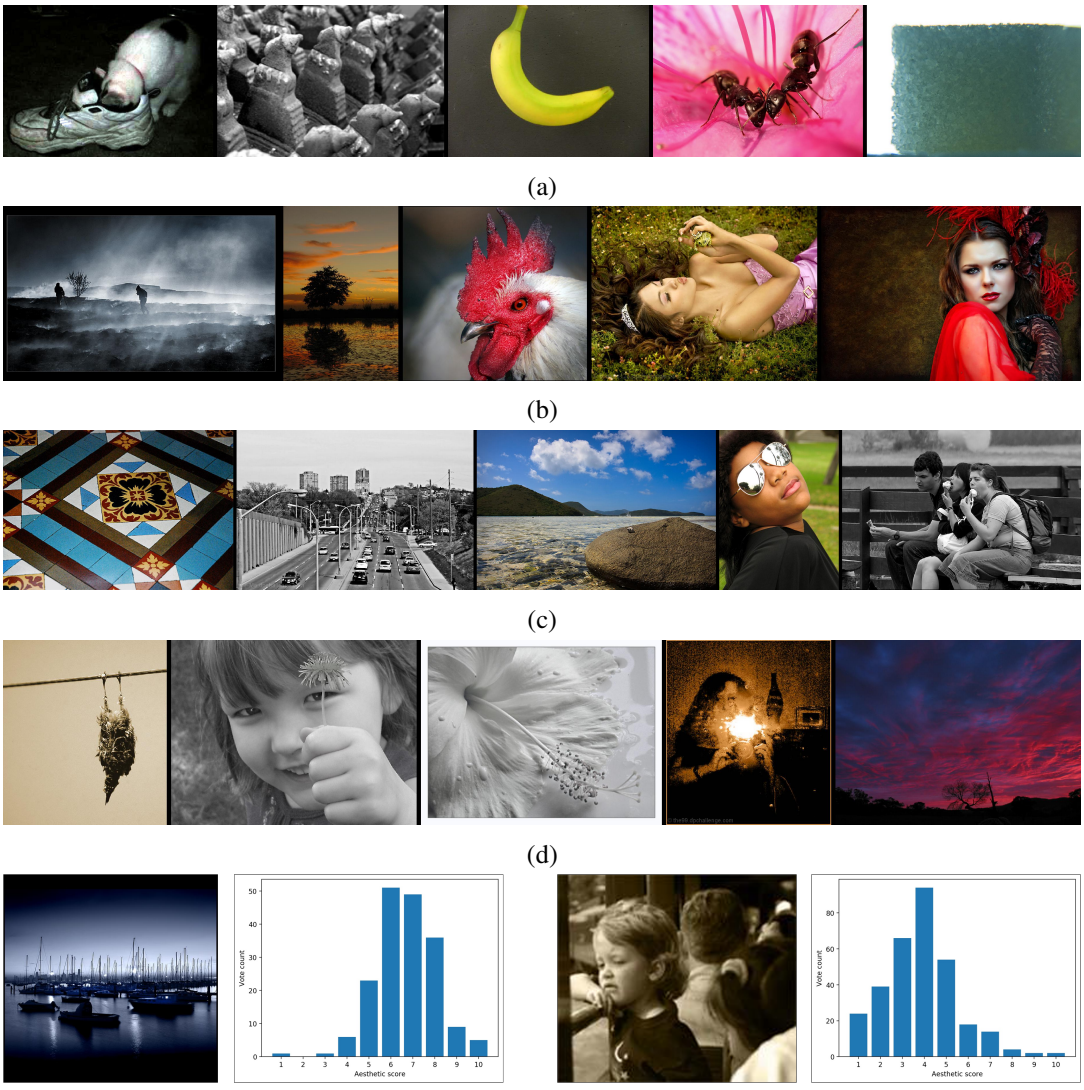


Figure 2.2: Figure showing example images from the AVA dataset that demonstrate the kind of images are used for learning aesthetic functions. We shows examples of images with (a) low mean score, (b) large mean score, (c) low standard deviation in user ratings, and (d) large standard deviation in user ratings. Part (e) show examples of high and low quality images (left and right respectively) along with their distribution of vote counts (which can be used to compute various statistics such as mean)

Flickr

Flickr is a photo-sharing platform offering a wide variety of options for editing, organizing and publishing pictures online with a vibrant community of amateur and professional photographers alike. The users can interact in a variety of ways: by following other users' photostreams, congregating in Flickr Groups (often centered around common themes such as Nature photography), post comments on photographs or make a photo a "favorite" of theirs (similar to the Like button on Facebook). Additionally, Flickr introduced a metric called "Interestingness" which is computed for each user based on their popularity, number of favorites in receives, viewing patterns of the photo and others.

DPChallenge

DPChallenge.com gathers together a community of photographers of all skill levels, amateurs and professionals alike, who participate theme-based photography contests. Owing to its popularity, the photographs on the website are rated by a large number of people (greater than one hundred ratings per photograph) on a ten point scale and with implicit labels on the photographs provided by the contest themes, serving as a great resource for mining labeled data.

Photo.net

Photo.net was originally started at MIT in order to promote research in online communities. The users can interact by sharing, rating and commenting on the photos. Each of the photographs can be rated based on two metrics: aesthetics and originality on seven point scales. A subset of approximately twenty thousand images can be found on the website of Rittendra Datta, one of the authors of the first aesthetics papers.

Terragalleria

Terragalleria is a collection of over thirty-five thousand travel photographs taken by a single person. It includes a large collection on Nature photographs (e.g. US National Parks) that can be rated by the viewers on the scale from one to ten. All of the photographs are available here.

Aesthetic Visual Analysis (AVA) dataset.

The AVA dataset is a large collection (more than 255,000 images) collected from the DPChallenge website augmented with **aesthetic** (distribution of user aesthetic ratings) , **semantic** (sixty-six diverse textual tags describing the semantic meaning) and **style** annotations (labels describing various aspects and rules of thumb of photography: complementary colors, rule of thirds, etc.). The full dataset can be found here.

CUHKPQ

CUHKPQ was collected at the Chinese University of Hong Kong with the aim of assessing photo quality. It contains more that seventeen thousand images divided into seven semantic categories with binary labels indicating high or low quality. However, because it was created by blending high quality professional photographs with low quality images from college students, it is thought not to necessarily be representative of the real difference between high and low quality picture. Dataset itself can be found at the following link.

The MIRFLICKR Retrieval Evaluation

The MIRFLICKR dataset (found here) contains a collection of twenty-five thousand images from Flickr aiming to be **open** (released under the CC License), **practical** (additional metadata is provided in a easily accessible manner), and **interesting** (only images with high interestingess measure are provided). Additionally, a second version of this dataset was released containing a total of one million images and additional content-based descriptors.

One Hundred Million Creative Flickr Images (OHMCFI)

The team at Yahoo Research collected a dataset consisting of 99.3 million Flickr images and 0.7 million videos and released them under the Creative Commons licensing. Each of the pictures come with the following features pre-computed available on the AWS: SIFT, GIST, Auto Color Correlogram, Gabor Features, CEDD, Color Layout, Edge Histogram, FCTH, Fuzzy Opponent Histogram, Joint Histogram, Kaidi Features, MFCC, SACC_Pitch,and Tonality. In order to request the access to the full dataset, submit a request at the following website.

Hidden Beauty of Flick Pictures

This dataset was collected as part of an effort to surface the ”hidden gems” among the pictures that have very low popularity / interestingness as measured on Flickr. Approximately 15,000 images were chosen from the sample of 9M images from the larger OHMCFI database. Although this database is not as large as the previously mentioned AVA database, it is a good starting point for aesthetics inference investigation. This is due to the fact that it is the largest database that aimed to ensure a controlled collection of the image labels. Although the labels were collected via the CrowdFlower crowdsourcing platform, each of the images was labeled by at least five different people (each evaluator having a top track record on the platform), with quality control in place, and the ratings for the pictures were clearly explained. Additionally, in order to justify the creation of the dataset, the authors benchmark the performance of the dataset against other ranking strategies and show that it outperforms all of the tested methods.

2.2.2 Handcrafted-Features

Studying computational aesthetics using a computational approach [20]

Datte et al. [20] was the first paper to tackle to problem of analyzing image aesthetics using computational techniques. They designed a set of features that would either align with principles of photography and guidelines in the literature regarding psychology of aesthetics. Although the images used as data (collected from Photo.net) were rated both on their originality and perceived aesthetics, the study limited itself to the examination of only aesthetics scores due to strong positive correlation with originality. Because of the role color tones and saturation play in photography and color psychology, the images were converted and processed in the HSV color space. The hand-crafted features were designed using the following principles: rules of thumb in photography, common intuition and observed trends in ratings. In the end, a set of 56 carefully chosen features was used to study the patterns leading to variations in aesthetic ratings. The features used in the study describe information about exposure, color, quality and composition of the image. Using all of the 56 features, they were able to achieve an overall 85.9% accuracy in predicting the high / low aesthetic value of the images and thus showing possibility to study aesthetics from the point of view of statistical learning.

Design of high-level features for photo quality assessment [47]

Aiming to distinguish between high quality ”professional photographs” and low quality ”snapshots”, Ke et al. present a top-down approach for constructing high level semantic features for photo quality assessment [47]. First the authors aimed to understand the perceptual criteria that people use to rate photos and then they distilled them to concrete measures usable as features for a machine learning algorithm.

After consultation of professional photography books and interviews with photographers and amateurs alike, the three distinguishing factors between the low / high quality photos come out to be:

- **Simplicity** simplicity in the sense that it is obvious what one should be looking at. In order to separate the subject and the background, the photographer can use the following techniques: *blur the background*, *choosing complementary colors for the subject and the background* and *lightning contrast between the subject and the background*.

- **Realism** - professionals use various techniques to snap atypical photos, that almost look surreal. Since photographers often take preparation, e.g. in choosing the time of the picture and camera settings, the color palette or subject matter are likely to be much different from

snapshots.

- **Basic Techniques** - which might indicate the quality of the photo such as blurry image or contrast.

Once Ke et al. established understanding of what distinguishes professional photographs, they designed features to measure the high level perceptual factors. Simplicity is measured by computing the Spatial Distribution of Edges to understand the presence of clearly defined subject. Hue Count measured the number of unique hues in the image. Ke et al. inspected the frequency content of the image, since whole-image blur is indicative of low quality of the photograph and thus indicates "bad technique". Since professionals are very adept at using color to highlight the subject, the color distribution of each image is compared to that of its k nearest neighbors. Finally, authors proposed low level features that capture contrast of the photo and its average brightness.

Learning the consensus on visual quality for next-generation image management [21]

Datta et al. proposed a novel architecture of rating images, motivated by the need to reliably retrieve high and low quality images. They proposed to exploit the available number of user ratings to estimate the consensus or degree of confidence in the estimated image quality score. Using the same features presented in [20], they trained a weighted Linear Least Squares Regressor and a Naive Bayes' classifier to jointly predict the quality of the picture.

Photo and Video Quality Evaluation: Focusing on the Subject [72]

All of the previous approaches used feature extraction techniques on the whole image. However as the rules of composition dictate, it is important to highlight the subject of the photograph in our image. Luo et al. recognized this and detected the subject by using blur detection to distinguish clear / blurry regions. From the detected region, they extracted a variety of features including clarity contrast, lighting contrast, complexity, composition and color harmony. They applied the same techniques to estimating the quality of the video and define two additional features to quantify motion stability and subject presence in the video. Using these subject-focused features showed superior performance compared to [47] in Web Image Ranking, and Photo and Video Quality Assessment.

Sensation-based Photo Cropping

Automated image enhancement is of immense interest, similarly to aesthetics inference, especially because of the large collection of images each of us possess. Photo cropping, a technique in which we select a subset of the image, has mostly been accomplished by selecting regions around either

humans, by using face detectors in photographs, or the main subject as determined by saliency detectors. Nishiyama et al. took a data-driven approach to this problem [83]. By using SVM quality detector trained on data from DPChallenge and Photo.net (a problem very similar to aesthetics inference), they cropped the photos by generating several possible candidates and picking ones with the highest quality score. To train the quality classifier, a set of subject regions is extracted from the saliency map. Each region is described by an edge, color and blur histogram. Each feature is used to train a probabilistic SVM to estimate whether they are of high / low quality and in the end serve as a mid-level feature for an SVM which will determine the overall quality of the image. A study of 30 users showed the success of the technique.

Saliency-enhanced image aesthetics class prediction

Many of the rules of photography, e.g. rule of thirds, tell us ways to manipulate the subject of our image to improve the appeal of the photograph. Recognizing the importance of subjects, Wong et al. used saliency enhanced segmentation to estimate the regions of the photograph that contain the subject [115]. They used a variety of features describing texture, blur, brightness, saturation and color to describe the whole image, subjects or the contrast between the subject and the background. Using feature selection methods to pick the top performing features (primarily of the image and the subject), Wong et al. were able to outperform previous methods by $> 5\%$.

Automatic aesthetic value assessment in photographic images

Jiang et al. [41] presented a framework for estimating a continuous ($0 \leq x \leq 100$) and discrete aesthetic value ($x \in \{1, 2, 3, 4, 5\}$) of image. To evaluate the framework, the authors used a previously collected dataset of more than 450 images each ranked by 30 users from their earlier work on understanding the aesthetics of consumer images [12]. Cerosaletti et al. [12] analysis of the dataset in the analysis of variance (ANOVA) factors as well as the artistic characteristics revealed attributes that characterize pleasing images, however mainly the fact that degree of aesthetics and technical image quality are highly correlated.

Jiang et al. [41] extracted from each image a set of visual features explored in previous studies [20, 47]. In order to estimate the continuous aesthetic rankings, they adapted the RankBoost algorithm. Given a dataset $\{x_1, \ldots, x_n\}$, with each x_i described by a set of ranking features $[f_1(x_i), \ldots, f_m(x_i)]$, RankBoost algorithm learns a ranking function H, which gives us a linear ordering on x_i's. By using an SVM as a weak learner in the Diff-RankBoost (DRB) algorithm, they learned to predict the relative aesthetic score between two images x_n and x_m, i.e. $H(x_n) - H(x_m)$. In order to regress the aesthetic value, the DRB algorithm is used to create a set of features $f_1(x_i), \ldots, f_n(x_i)$ for each image x_i, where $f_j(x_i) = H(x_n) - H(x_m)$. These image features are then fed into a Support Vector Regression Machine to predict the actual targets y_n. Secondly,

the authors quantized the fine-grained aesthetic scores into five categories and trained a five-class SVM on the image features to predict these classes.

The role of image composition in image aesthetics

Image composition is one of the main determinants of image aesthetics as note by [47, 93]. Although often extracted features consider parts of image composition, Obrador et al. [85] were the first to do a detailed study of image composition as determinant of computational aesthetics. They described a total of 55 features that consider the image simplicity, region relevance, layout appeal and visual balance. Individually the features came short in predicting aesthetics compared to previously published methods. This might be due to the very specific nature of the features not encompassing all aspects of aesthetics. However the potential of the features became apparent when combined with previous features, which significantly improved aesthetics inference.

A Framework for Photo-quality Assessment and Enhancement Based on Visual Aesthetics

Bhattacharya et al. presented a framework for inferring and altering the aesthetics of the photograph [9]. The scope of the paper was limited to studying only two types of scenes: one with a single main subject and one without (e.g. landscapes and seascapes). For this study, Bhattacharya et al. assembled a dataset of 632 images and conducted user survey to rank each of the images on a five point scale. In order to learn aesthetic preference, they proposed two types of features for the different scenes that capture the composition information about the image. For single subject photos, they estimated the region of the image and estimate its center of mass or "visual attention center" using semantic segmentation. To characterize the image, they created a 4 dimensional vector with distances to the four focus points, intersection of the horizontal and vertical lines that divide the image into nine parts. For images without subject, they described each image by ratios of vertical extents of the sky and the support using the same semantic segmentation techniques. According to the rules of the composition, they should be as close to the golden ratio as possible. They learned SVR classifiers for each of the categories. In order to improve the aesthetics of the images, they proposed to relocate the subject of the image or change the ratio of sky / support by extending / cropping their regions, which resulted in improvement of estimated aesthetics in 73% of the time.

Learning to Photograph

Cheng et al. used a large set of approximately 10^5 crawled images to automatically learn the ideal rules for image quality and suggest the ideal view the photograph should take [17]. Each image was divided into a set of atomic regions using graph-based image segmentation, with each region characterized by a feature vector composed of Color Harmony and HOG texture features.

Using K-means clustering, they created a visual vocabulary of 1000 words from all of the patches extracted from images and then described each image by a Bag-of-Visual-Words. Furthermore, the images were divided into 100 sub-topics using K-means clustering and a separate probabilistic model, that models both the presence of visual words and their co-presence, was learned for each sub-topic. Using a group of fifty human subjects, they validated their model and showed its success compared to previous re-targeting models which utilized visual saliency.

High-Level Visual Attributes for predicting Visual Aesthetics

Dhar et al. developed techniques for estimating high-level describable features (kinds of characteristics that a human might use to describe an image) [24]. They fall intro three distinctive categories:

- **compositional attributes** - characteristic related to the layout of an image that indicate how closely an image follows photographic rules of composition
- **content attributes** - characteristics related to the presence of specific objects or categories of objects including faces, animals, and scene types.
- **sky-illumination attributes** - characteristics of the natural illumination present in the photograph.

For Dhat et al., describability of the attributes was essential, as group of people can be queried regarding the presence and absence of such attributes. This data was then used to train classifiers to predict these attributes and estimate aesthetic value / interestingness of images. The research built upon work on face recognition, where face attributes, e.g. race and gender, were shown to improve facial recognition results. One of the main contributions of the paper was showing that by training classifier to estimate the describable attributes and then using these attributes as features can significantly improve the prediction of the aesthetics and interestingness scores.

A total of 26 classifiers indicating the presence of above features was trained on hand-labeled data collected from: Flickr, Photo.net and Animals on the Web dataset. In order to estimate the aesthetics and interestingness, additional sets of sixteen and forty thousand images were collected on DPChallenge.com and Flickr, respectively. The top 10 % of images were labeled as high aesthetic quality / interestingness value and bottom 10 % of the images were used to denote the negative examples. Dhar et al. trained an SVM classifier using the 26 describable features and demonstrated effectiveness in measuring both. In each case, the SVM classifier trained on high-level features outperformed a baseline Naive Bayes classifier by Ke et al. trained on low-level features. The classifier performance was further improved by combining both high-level attributes and low-level features.

Aesthetic Quality Classification of Photographs Based on Color Harmony [82]

Previous papers recognized the importance of color in aesthetics inference. This was especially due to the evidence from study of psychology and art theory, which showed that color often induces different emotion in people. Nishiyama et at. [82] pointed out that previous papers used rather simple descriptors of color, e.g. average values of the RGB channels, or color histogram. Nishiyama et al. proposed a new set of features, 'bag-of-color-patterns', which aimed to characterize the color harmony of the photograph.

Color harmony is a property certain combinations of color are said to have if they together have an aesthetically pleasing effect on the observer. Otherwise, non-harmonious combination of color would not engage the observer or make them look away from the picture, in the case of chaotic colors [79]. Color Harmony is mainly discussed by two models: The Moon-Spencer model and the Matsuda methods. These models have been used in variety of applications to design harmonious color combinations, e.g. marketing campaigns, website design, clothing pattern design. They cannot however be used to describe the color harmony of a picture, whose spatial color pattern is much more complicated than that of a simple dress design.

In order to describe the harmony of the image, the authors proposed to sample smaller patches of the image where each region ends up having a simpler combination of colors. Then one can use color harmony models to describe each of the patches, which are then combined to a descriptor of color harmony in the image. In their algorithm, authors sampled the image on a uniform grid, diving the sampled regions into uniform and ones with color edges. Each region was then described by a color histogram in the CIE LCH color space. To create the bag-of-color-patches features, a large number of local patches was sampled from the images in the training set and codebooks for the regions with / without color edges were created using k-means clustering. Lastly, each image was divided into several larger regions, each described with a histogram of local image patches which are then concatenated to create a representation of the whole image. Combining the bag-of-color-patches histogram, with blur, saliency and edge features showed superior performance of such methods outperforming the competing models by a large margin.

Content aware aesthetics

The purpose behind or the type of the photograph the author is trying to take is going dictate many of the choices of the photographer: if they aim to capture a particular subject, e.g. an animal, a plant or an insect, they are likely to choose a blurry background to focus our attention. Otherwise, if they take the picture of a person, we will naturally be drawn towards a human face. This is indicative of the fact that different photo categories would require distinctive mix of features to recognize the its quality.

Tang et al. were one of the first works to explicitly consider photo content in [71, 100]. They collected a dataset consisting of seven photo categories and computed two types of features: global (computed on the whole image), and local (computed on the subject of the image).

To estimate the subject of the image, they considered three different subject area extraction techniques for the different categories. The subject area for:

- ”animal”, ”static”, ”plant” and ”night” categories was extracted by estimating the blur in the image.
- ”architecture” and ”landscape” was estimated by extracting vertical standing objects from a previously published scene segmentation algorithm.
- ”human” was estimated using face / human detection algorithm.

In order to evaluate the subject areas of the images, the proposed the following regional features:

1. Dark Channel Feature - average normalized dark channel value (described in [32]).
2. Clarity Contrast - aims to capture the sharpness of the subject areas by comparing the frequency content in the subject and the whole image.
3. Lighting Contrast - compares the average lighting between the subject and the background.
4. Composition Geometry - measures the Rule of Thirds by computing the minimum distance to one of the four image intersections (as defined by the two horizontal / vertical lines diving image into nine regions).
5. Complexity Features - aims to capture the complexity of the image by counting the superpixels the background / subject is segmented into.
6. Human Based Features - aims to capture the quality of portraits by considering the ratio of face area in the image, amount of shadow in the faces, average lighting of the faces, and their clarity.

To capture the information about the image, the authors extracted:

- Hue Composition Feature - which aims to capture the color harmony of the image by considering where the majority of hues values cluster on the color wheel.
- Scene Composition - by using the Hough Transform to extract the horizontal and vertical lines in the image, they aim to capture the average position of orientation of these lines.

Tang et al. treated the scene categories as ground truth. To solve the problem of estimating the image category, they proposed the following method:

- compute the Edge Orientation Histograms, HOG and GIST features.
- retrieve the top 100 high / low quality nearest neighbors from the training data.
- using the training labels, estimate which of the subject extraction techniques is to be used
- train a linear SVM based on the retrieved training samples to estimate the class of the test sample.

Tang et al. noted their features outperformed their benchmarks, though they did not compare their results to any features / algorithms that were published recently in their time frame, e.g. the color harmony features described in the previous section.

Obrador et al. [84] took a similar to approach and introduced their own dataset collected at DPchallenge.com. They computed three categories of features:

- Simplicity features - quantified by measuring various statistics about the regions of graph-based segmentation algorithm.
- Global features - 38 low level features capturing information about the luminance, contrast, colorfulness, color harmony, and composition (e.g. rule of thirds) of the image.
- Contrasting features - by using sharpness, luminance, chroma, relevance and saliency, they create five binary maps that classify the image into subject / background and compute various low level features that measure sharpness, exposure, chroma and saliency of the regions.

Contrary to most of the models, Obrador et al. trained a SVM regressor to predict the real valued scores, instead of binarizing the scores. After computing all of the features for the training images, they used feature selection methods to obtain the set of best performing features for each of the categories and demonstrated the improvement in aesthetics score prediction for category specific models as opposed to the generic model.

2.2.3 Local Features

Zhang et al. [120] proposed a graph-based probabilistic approach for aesthetics inference which aimed to capture both local and global image information. In their approach, they segmented the image into several 'atomic' regions using unsupervised fuzzy clustering, forming a graph $\mathcal{G} = (V, E)$. V is the set of vertices each corresponding to the atomic regions of the image end E is the edge set representing the adjacent regions in the image. Each region was represented by a three sets of visual features: HOG (128-d), Color moment (9-d) and visual saliency features (64-d). For each image, a set of five hundred graphlets, connected induced subgraphs, was sampled with each of maximum size T (to be specified by the user). A t-vertex graphlet was represented

by four matrices: M_R^C, M_R^T, M_R^S and M_S. $M_R^C \in \mathbb{R}^{t\times 9}$, where each row is a 9-d color vector for each of the regions in the graphlet (M_R^C, M_R^C are defined identically). M_S aims to capture the local structure of the connections between the t regions, very similar to a adjacency matrix. Then all of these features were concatenated into three matrices $\{M^C, M^T, M^S\}$, where $M^C = [M_R^C, M_S]$.

The authors proposed a manifold embedding, which was used to transform the different sized matrices into fixed d-dimensional vectors and encodes the global spatial layout into the graphlets as well. Once all of the graphlets (for test and train images) were transformed into post-embedding graphlets, they were used in a probabilistic graphical model to compute $\gamma(I^*) = p(I*|I^1, \ldots, I^H)$. As the authors noted: $\gamma(I^*)$ can roughly be interpreted as the "amount of graphlets that can be transferred from the training photos into the test one". They demonstrated the efficacy of the algorithm by comparing the algorithm on three datasets against five different feature extraction methods and achieving state of the art results.

2.2.4 High-level Features

By using the CNNs, one can automatically discover the features as opposed to using :

- handcrafted features, which are merely approximation to photographic rules,
- generic features.

One of the main advantages of using hand crafted or generic-features is that they compute a fixed representation of the picture, the same number of features for each image. Thus it is easy to take the features and feed them into a learning algorithm. However, applying CNNs can at times prove to be tricky, since NNs take a fixed input as well and images often come in various aspect rations and sizes.

Lu et al. [68] were among the first to tackle the problem of aesthetics inference that solely uses CNNs. They conducted a thorough study of several network architectures and experimented with constructing a multiple column network with varying inputs. As different photographic rules (e.g. rule of thirds or color harmony) consider properties on different scales of the image, authors used both the global and the local view (random fixed crops) of the image to train the networks. To get around the problem of varying image sizes, following image transformations to fix the size of the image to $s \times s \times 3$ were proposed:

- **Center-crop** - resize the shorted side of the image to a fixed size s and take the "center" s pixels to form the image
- **Warp** - anisotropically resize each of the sides to a fixed size s

- **Padding** - resize the longer side of the image to a fixed size s and padding the rest of the image with zeros.

Since the global and local details of the image are important, the authors proposed a Double Column CNN (DCNN), where one of the columns accepts a global-view of the image and the other a local-view. It was essentially a network consisting of two separate Single Column CNN (SCNN) whose outputs were combined to produce a single score. Because of the large intra-class variation in aesthetics scores, Lu et al. proposed to use higher-level style labels, that are available for a subset of images present in AVA dataset, as additional features. They trained an additional Style SCNN to recognize the various style labels (e.g. complementary colors, motion blur or the rule of thirds). The trained network was then used to extract the Style features for the rest of the images in the AVA dataset. These features were then concatenated with the features computed by the DCNN network and used to determine the final aesthetics score. A very interesting detail arose when looking at the images correctly classified by the DCNN and incorrectly by SCNN: when the input was a local-view of the image, it often contained a large subject and similarly when the input was a global-view, the image often contained specific texture likely to be better seen on the local-view.

Wang et al. took inspiration from Neuroaethetics and Neuroscience of Vision to propose a novel architecture that aimed to tackle the problem of binary classification and the distribution of aesthetics scores [108]. They proposed a model called the Brain-Inspired Deep Network (BDN) composed of two parts:

- Parallel Pathways layer
- High-level Synthesis Network

The *Parallel Pathways layer* was inspired by the parallel pathway processing of the human cortex, which decomposes the visual scene into several representations that encode information such as intensity and edge information in the image. In this layer they converted the RGB data into the HSV format and used each H, S, V as one of the parallel representations for the image. As was shown previously in [24], high-level attributes are successful in augmenting aesthetics prediction if used as mid-level features. Therefore Wang et al. decided to train fourteen fully-convolutional networks (FCN) trained in a supervised ways to predict the fourteen binary style labels available with the AVA dataset. The activation of the mid-level convolutional networks for each of the fourteen style label were used in parallel as features for the high level synthesis network (thus virtually decomposing image into several representations encoding different information). The synthesis network, also a FCN, was used to predict the binary high / low aesthetic label as well as the distribution of ratings (by predicting the mean and deviation of a Gaussian distribution trained by minimizing the Kullback-Leibler divergence).

Guo et al. [29] proposed to use paralleled Deep CNN (PDCNN) architecture. They utilized an architecture similar to the AlexNet that won the 2012 ImageNet Competition [55]. Since the CNNs are prone to both over- and under-fitting, they proposed to use PDCNN to control the complexity of the system by stacking n columns in parallel - n-PDCNN. As they showed, the performance of the network network improved when they stacked up to three networks in parallel. Thereafter, with more then four paralleled inputs the performance started to degrade.

Kong et al. [53] proposed AlexNet-inspired architecture to predict various image attributes. First, they created a simple regression network to predict aesthetic rating by minimizing the Euclidean loss. Subsequently, they adopted a Siamese Network architecture [11] to jointly optimize the network to both predict an aesthetic score and a relative ranking of the two images. Similar to [108], Kong et al. predicted the aesthetic attributes of images augmenting the network with an auxiliary task of predicting attributes from the same activations that were used to predict the aesthetic score. Lastly, the network was used to predict image categories.

2.2.5 Related Work

What makes images pleasing ?

Marchesotti et al. set out to discover sets of textual image attributes that could be used a mid-level features for various tasks, including image ranking or aesthetics prediction [70, 76]. They used the AVA dataset [80], which along with images, their aesthetic scores and style labels, contains comments associated with each of the images. The comments were used to construct Bag-of-Words feature vectors (using tf-idf (term frequency-inverse document frequency) feature representation for the words) for each of the images and used the elastic net model (a linear combination of l_1 and l_2 regularization) to discover attributes most predictive of beautiful / ugly pictures. They used both unigrams and bigrams, and found that bigrams were much more suitable as attributes since unigrams often yielded ambiguous descriptions (“not”, “out”, , “don’t”). They picked the top 1500 ugly and beautiful attributes and used spectral clustering to reduce the number of total labels to 200 (using the Levensthein word similarity measure to gauge the similarity of the second word, since it is the one thought to contain the actual semantics with the first word bearing the polarity of the attribute, e.g. great lighting). Then they used images with the discovered attributes to train one-vs-all classifier to use them to describe images using features that are actually human interpretable.

Majority of the work done is formulated as either a classification or regression problem with the goal of predicting a single aesthetic judgement / score that tells us how appealing the image is. However it often does not explain what makes the image appealing or not. When designing most of the aesthetic inference algorithms, many of the features (numbers that we compute to represent various information about the image) are solely chosen for their ability to improve the algorithm

(in terms of classification accuracy or RMSE score). Aydin et al. [6] set out to remedy this issue and defined a set of image attributes one would quantify and would be related to photographic principles, thus help us guide in taking better pictures and provide an objective basis for comparing their aesthetic appeal.

In order to choose what photographic principles to capture in their aesthetic attributes, the authors chose the following criteria that the attributes should satisfy:

- **Generality** - e.g sharpness, which can be used to judge every image versus an attribute such as facial expression
- **Relation to photographic rules**
- **Clear definition**

The attributes that were in the end chosen by the authors were:

1. **Sharpness** - aims to capture the level of detail seen in the in-focus region.
2. **Depth** - aims to capture the impression of ”depth” in the picture, which can for example be created by blurring the background of the image.
3. **Clarity** aims to capture whether the photograph or image has a clear principal idea, topic, or center of interest.
4. **Tone** is related to the magnitude of the global lightness differences.
5. **Colorfulness** - as discussed previously, pictures with a vibrant color palette are often considered to be more aesthetically pleasing.

By defining, computing and calibrating (making sure the actual output values have equal ranges), the authors presented a framework that computes an aesthetic signature of the image, comprising of measures of visual image attributes that relate to specific photographic principles. They demonstrated the performance of the ratings by exploring several possible applications such as:

- Automated Aesthetic Analysis
- Photo editing evaluation
- HDR Tone Mapping Evaluation
- Multi-Scale Contrast Editing

Image ranking and retrieval

Image ranking is a problem of great importance, especially to image search engines which aim to index the content of the web and then serve it to the user at their will. When retrieving the images for a given query, the images that are shown should take into account three main aspect: semantic relevance, quality and variety of images [44]. The quality of the images can be quantified by various metrics: popularity among images on a social networking website like Flickr or aesthetics value computed using an algorithm.

Multidimensional Image Value Assessment and Rating for Automated Albuming and Retrieval [66]

Loui et al. [66] proposed a multidimensional image value index (IVI) to be used by an automated system for raking of images. It comprises a signature of a photo that captures various higher level information about it: its quality, aesthetics as well as relevant semantic value to the user. M-IVI comprises of five IVI values: Technical, Aesthetic, Social, Event and Usage. The authors proposed to learn the Technical IVI in a data-driven way. On the contrary, the social IVI was determined in more heuristic / manual definition of some of the values, since it tries to capture the personal relevance of the people in the photographs to the user. Loui et al. demonstrated the feasibility of implementing the Social and Technical IVI, however more work would have to be put in to adequately define rest of the values. This was due to the fact since the meaning they tried to capture either requires a lot of additional user input (not always desired or easily available) or implementation of a complicated system that combines face detection, metadata, etc.

Marchesotti et al. [81] described an algorithm that aims to augment the image retrieval system by ranking the images not only according to their semantic relevance to the query, but also aiming to ensure that the images are aesthetically pleasing. They first introduced a previously published algorithm from an earlier paper that learned a joint model that will take into account both semantics and aesthetics. [81] however demonstrated that learning separate models for semantics and aesthetics provides much better performance in ranking. They proposed the following two models:

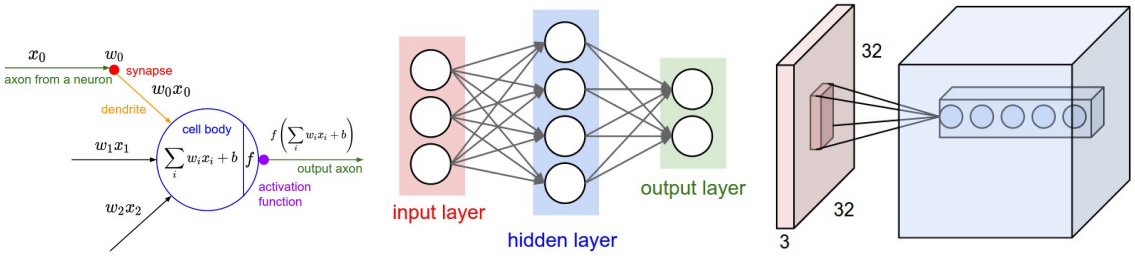
- **Independent Ranking Model** - assumes that aesthetics and semantics are separate given image:

$$p(y, z|x) = p(y|x)p(z|x)$$

- **Depender Ranking Model** - assumes that aesthetics is going to be dependent on the semantics of the image:

$$p(y, z|x) = p(y|x)p(z|y, x)$$

As they demonstrated in the metrics, both of these models outperformed the original model, which aimed tries to learn a joint model.



(a) Artificial neuron (b) Elementary neural network (c) Local connectivity of a neuron.

Figure 2.3: Figures showing (a) model of the neuron used in neural networks, (b) architecture of an elementary neural network with a single hidden layer, and (c) the local connectivity of a neuron in a convolutional layer. Source: https://cs231n.github.io/

2.3 Neural Networks and Deep Learning

In this section we briefly discuss neural networks, which currently dominate the approaches in several areas of computer vision, e.g. object detection and classification, face detection or speech recognition. We discuss backpropagation, a method of optimizing weights of a neural network, a basic model of a neural network, and lastly ranking optimization that is utilized in Chapters 5 and 6 of this thesis.

2.3.1 Backpropagation

Backpropagation is a particular method of computing the gradients of the loss / error function with respect to the parameters, using the multivariable chain rule, by considering the loss function as being a composition of several nonlinear functions. For example, consider the function $f(g(x(t), y(t)))$. In order to backpropagate the gradient, we first find $\frac{\partial f}{\partial g}$. Then we have $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial x}$ and $\frac{\partial f}{\partial y} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial y}$. In the end, we have $\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t}$. Similarly, the gradient of the loss function with respect to weights is calculated by backpropagating the error through the different layers of the neural network. Once the gradient is computed with respect to all of the parameters, stochastic gradient descent [114] (or a variation of it) is used to update their value. The value of parameter w at step $n + 1$ would thus be updated as follows:

$$w^{(n+1)} = w^{(n)} - \alpha \frac{\partial L}{\partial w}.$$

2.3.2 Neuron function

Figure 2.3a shows the model of an artificial neuron that approximates the function of real neuron with all of its parts: synapse, dendrite, cell body and axon. The inputs, $x_1, \ldots, x_n$ into neurons are represented as signals traveling through dendrites, whose effect is modulated by the synaptic weight w_i. The effect from all of the weighted inputs is then summed together and biased with a bias term b. The sum is then transformed via a nonlinear function (paralleling the neuronal firing of a neuron) and the output travels along an "axon" and serves as an input into further neurons. As can be seen in Figure 2.3a, the output of the neuron can be written as:

$$\begin{aligned} y &= f(\sum_i w_i x_i + b) \\ &= f(\mathbf{x}^T \mathbf{w}). \end{aligned}$$

In the above equation, the bias term b was absorbed as a weight w_0 with $x_0 = 1$

2.3.3 Neural Network

This section describes a basics of neural networks, and describes equations for a 2 layers feed-forward neural network for classification, using the softmax loss function, seen in Figure 2.3b, which can then be extended into more complicated models, e.g. the convolutional neural network. The following variables be of interest:

- N - number of inputs processed at the same time
- D - number of input dimensions
- H - number of hidden units (neurons)
- K - number of classes for classification

Additional notation has to be introduced. Since each of the neurons in the hidden layer is fully connected to all of the inputs, each neuron will have a separate synaptic weight modulating the input differently. Thus let $w_{i,j}$ represent the weight modulating the i^{th} input for the j^{th} hidden neuron. We can store all of the weight conveniently in a weight matrix $\mathbf{W}$:

$$\mathbf{W} = \left[w_{i,j}\right]$$

In our network, we will need two weighting matrices: $\mathbf{W}^{(1)}$ transforming the inputs to activation of hidden layer neurons and $\mathbf{W}^{(2)}$ transforming neuron outputs to the K class scores. The inputs are going to be stored in matrix $\mathbf{X}$, where $x_i^{(n)}$ is the i^{th} input feature of the n^{th} example.

$$\mathbf{X} = \left[x_i^{(n)} \right]$$

Therefore starting with an example matrix $\mathbf{X}$ and weight matrix $\mathbf{W}^{(1)}$, the transformed outputs of the neurons are found as follows:

$$\mathbf{H} = f\left(\mathbf{X} \cdot \mathbf{W}^{(1)}\right),$$

where $\cdot$ represent matrix multiplication and $f(\cdot)$ represented element-wise application of the rectified linear unit (ReLU) nonlinearity [55]:

$$f(x) = \begin{cases} x & \text{, if x} > 0 \\ 0 & \text{, otherwise} \end{cases}$$

Furthermore, $\mathbf{K}$, the matrix of class scores for the all of the examples are calculated:

$$\mathbf{K} = \mathbf{H} \cdot \mathbf{W}^{(2)},$$

where each element $f_k^{(n)}$ represents the score indicating the belief that the n^{th} example belongs to k^{th} class. To make these notions more clear, we will convert the class scores into probabilities using the normalized exponential function also known as softmax [113]:

$$p_k^{(n)} = \frac{e^{f_k^{(n)}}}{\sum_i e^{f_i^{(n)}}}$$

Since we are performing supervised classification, each data point $\mathbf{x}^{(n)}$ is accompanied by a corresponding class label $y_n \in \{1, \ldots, K\}$. The loss function to be optimized for each example is the softmax loss function:

$$L_i = -\log(p_{y_i}^{(i)}),$$

and the total loss for all of the examples is defined as follows:

$$L = \underbrace{\frac{1}{N} \sum_i L_i}_{\text{Data Loss}} + \underbrace{\sum_i \sum_j \left(w_{i,j}^{(1)}\right)^2 + \sum_i \sum_j \left(w_{i,j}^{(2)}\right)^2}_{\text{Regularization Loss}}$$

Function L can be further viewed as a composite function of the network parameters, and thus we can use backpropagation to optimize the network weights to minimize its functional value to improve performance of the model. Using the chain rule, we can show that:

$$\begin{aligned}\frac{\partial L}{\partial f_k^{(n)}} &= \frac{1}{N}\frac{\partial L_n}{\partial f_k^{(n)}} = \frac{1}{N}\left(\frac{1}{p_k^{(n)}}\frac{\partial}{\partial f_k^{(n)}}(p_k^{(n)})\right) \\ &= \frac{1}{N}\left[p_k^{(n)} - 1(k = y_n)\right].\end{aligned}$$

Let ∂L represent the matrix of partial derivatives with respect to $f_k^{(n)}$. Furthermore, we would like to find $\frac{\partial L}{\partial \mathbf{W}^{(1)}}$ and $\frac{\partial L}{\partial \mathbf{W}^{(2)}}$. Note, that for $\frac{\partial L}{\partial \mathbf{W}^{(1)}}$, we will have to back-propagate the error through the hidden units as described in the previous section. Thus first, consider $\frac{\partial L}{\partial w_{i,j}^{(2)}}$ (temporarily omit $\lambda w_{i,j}^{(2)}$ from the regularization loss):

$$\begin{aligned}\frac{\partial L}{\partial w_{i,j}^{(2)}} &= \sum_n \frac{\partial L^{(n)}}{\partial w_{i,j}^{(2)}} = \sum_n \frac{\partial L^{(n)}}{\partial f_j^{(n)}}\frac{\partial f_j^{(n)}}{\partial w_{i,j}^{(2)}} \\ &= \sum_n \frac{\partial L^{(n)}}{\partial f_j^{(n)}} a_i^{(n)} = \mathbf{a}_i^T \cdot \partial L_j,\end{aligned}$$

where $\mathbf{a}_i$ is the column vector of i^{th} hidden layer neuron activations and ∂L_j is the j^{th} column of ∂L. Since $\frac{\partial L}{\partial w_{i,j}^{(2)}} = \mathbf{a}_i^T \cdot \partial L_j$, we can see that $\frac{\partial L}{\partial \mathbf{W}^{(2)}}$ simply is:

$$\frac{\partial L}{\partial \mathbf{W}^{(2)}} = \mathbf{H}^T \cdot \partial L + \lambda \mathbf{W}^{(2)}.$$

To propagate the error back to earlier layers, it is necessary to compute $\frac{\partial L}{\partial a_i^{(n)}}$:

$$\begin{aligned}\frac{\partial L}{\partial a_i^{(n)}} &= \sum_k \frac{\partial L}{\partial f_k^{(n)}}\frac{\partial f_k^{(n)}}{\partial a_i^{(n)}} = \sum_k \frac{\partial L}{\partial f_k^{(n)}} w_{i,k}^{(2)} \\ &= \partial L^{(n)} \cdot \left(\mathbf{w}_{i,}^{(2)}\right)^T,\end{aligned}$$

where $\partial L^{(n)}$ is the n^{th} row of ∂L and $\mathbf{w}_{i,}^{(2)}$ is the i^{th} row of $\mathbf{W}^{(2)}$. Form of $\frac{\partial L}{\partial a_i^{(n)}} = \partial L^{(n)} \cdot \left(\mathbf{w}_{i,}^{(2)}\right)^T$ hints at the fact that ∂H can simply be computer as:

$$\partial H = \partial L \cdot \left(\mathbf{W}^{(2)}\right)^T$$

To propagate the errors through the nonlinearity, we note that:

$$\frac{d}{dx}f(x) = \begin{cases} 1 & \text{, if x} > 0 \\ 0 & \text{, otherwise} \end{cases}$$

Thus, all the derivatives in ∂H for which the forward pass was less than or equal to zero, we set to zero. Now to propagate the errors to $\frac{\partial L}{\partial w_{i,j}^{(1)}}$, we simply note that this is equivalent problem to what was previously solved by treating ∂H as ∂L and $\mathbf{X}$ as $\mathbf{H}$, getting that:

$$\frac{\partial L}{\partial \mathbf{W}^{(1)}} = \mathbf{X}^T \cdot \partial H + \lambda \mathbf{W}^{(1)}.$$

In a multilayer network, the previous layer would likely be another hidden layer and its gradient would be obtained as:

$$\partial H^{(l-1)} = \partial H^{(l)} \cdot \mathbf{W}^T.$$

In the following section we describe convolutional neural networks, models popular for processing images, and a particular loss function for optimizing pair-wise ranking of examples.

Convolutional Neural Network (CNN)

CNN is very similar to a simple NN presented in the previous section, however it modifies its structure to accommodate for the assumption that the input is an image [55]. Thus its input and individual layers are no longer described by input and output vectors, but rather volumes, where each of the input/output stages has width, height and depth (illustrated in Figure 2.3c.). CNN further differs from the simple NN by the input regions of the individual neurons. In the simple NN, each neuron in the hidden layer was fully-connected to all of the inputs from the input / previous hidden layer. CNN takes its inspiration from the Visual Cortex and allows the neurons to be affected by a smaller spatial regions of the input as can be seen in Figure 2.3c. A particular characteristic of CNN is that it employs weight-sharing. This means that neurons in the same slice share the weights for the activations at the different spatial regions. This amounts to a significant reduction in the number of parameters and thus partially preventing over-fitting and improving generalization. By employing weight-sharing and sweeping the same set of weight through the whole input, the network essentially preforms a convolution of the input with the weight matrix. Because of this, the different sets of weight convolved with the input image are called convolution kernels or filters that the CNN learns.

2.3.4 Implicit learning of ranking functions

In this section, we describe the Hinge ranking loss function [16, 58, 110], which is used to learn a ranking function that aims to preserve the ground truth ranking of image pairs. Figure 2.4 shows

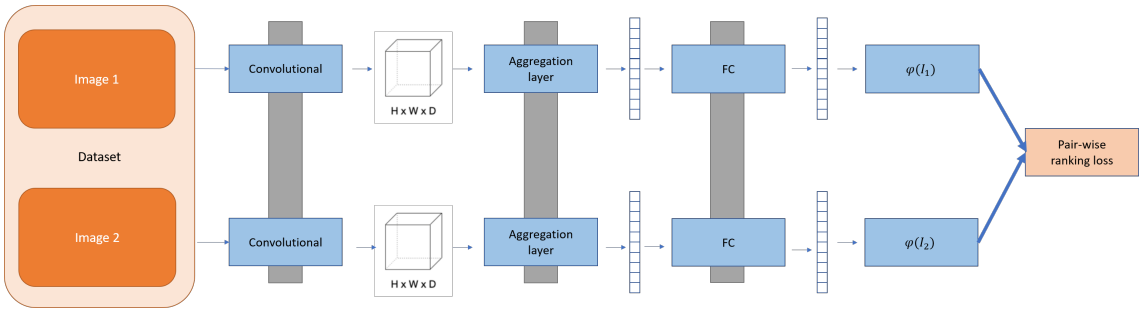


Figure 2.4: A schematic of a Siamese Network used to learn a ranking function via pair-wise learning.

a basic schematic for pair-wise ranking learning. The grey color in the figure indicates that the weights between the corresponding layers are shared.

Formally, the goal of pairwise ranking is to (implicitly) learn a ranking function

$$\phi : \mathbf{X} \rightarrow \mathcal{R},$$

which maps an image $I \in \mathbf{X}$ into a single value ϕ that ought to preserve the ground truth ranking between pairs of images. One such function, and one used in our work presented in Chapters 5 and 6 is the Hinge ranking loss, which takes the following form:

$$l_{rank} = \frac{1}{N}\sum_{i=1}^{N} L(I_i^1, I_i^2) = \frac{1}{N}\sum_{i=1}^{N} \max(0, \xi - \delta(y_i^1 \geq y_i^2)(\phi_i^1 - \phi_i^2)). \tag{2.1}$$

where $\delta(y_i^1 \geq y_i^2)$ is 1 if $y_i^1 \geq y_i^2$ otherwise it takes the value of -1, ξ specifies the margin parameter, and y_i^1 and y_i^2 are the ground truth composition scores for image 1 and image 2. Note that, if the ranking function is used by itself to optimize the Neural Network, one can write it in the following simple form:

$$l_{rank} = \frac{1}{N}\sum_{i=1}^{N} \max(0, 1 - \delta(y_i^1 \geq y_i^2)(\hat{\phi}_i^1 - \hat{\phi}_i^2)), \tag{2.2}$$

in which one factors out the margin parameter ξ and we get a transformation $\hat{\phi} = \frac{\phi}{\xi}$. In cases in which the function ϕ is jointly optimized by the ranking loss, and a regression loss, one should optimize the ξ by taking into consideration the scale of ground truth values.

To better understand this ranking criterion, consider the case in which we are given an image pair (I_i^1, I_i^2) and we know that $y_i^1 \geq y_i^2$. In such case, we have that $\delta(y_i^1 \geq y_i^2) = 1$, and the loss

function takes the following form:

$$L(I_i^1, I_i^2) = \max(0, \xi - (\phi_i^1 - \phi_i^2)). \tag{2.3}$$

Analyzing this function, we can see that we will incur a loss in case second argument of the max function is positive, i.e.

$$\xi - (\phi_i^1 - \phi_i^2)) > 0, \tag{2.4}$$

giving us

$$(\phi_i^1 - \phi_i^2)) < \xi, \tag{2.5}$$

i.e. telling us that the difference $\phi_i^1 - \phi_i^2$ is smaller that our desired parameter ξ.

2.4 Summary

In this chapter, we

- introduced a standard approach for tackling image aesthetics inference,
- discussed sources of data for aesthetic inference,
- discussed prior work on aesthetic inference,
- and discussed basics behind neural networks.

In many areas of computer vision, deep learning allowed for a significant improvement in performance in the given fields. Similarly, in computational aesthetics authors made the jump from hand-crafted features and more traditional approaches and started to utilize deep learning for predicting image aesthetic quality. Our first endeavor aimed to bridge this jump and try to understand whether there is still knowledge to be gained from more traditional approaches. As we will see in Chapter 4, traditional features can aid neural networks and features that capture color information, image quality, and photographic rules were the most helpful in improving the performance of deep learning algorithms.

With the increase in popularity in cell phone photography, we saw and explosion in the number of photos captured and shared that came from cell phone cameras. In 2017, 50 % of the registered users on Flickr used Smartphones for their photos, as compared to DSLR (33 %) and Point-and-Shoot cameras (12 %) [40]. Our work in Chapter 5 aimed to explore how well can one perform aesthetic ranking using constrained neural network models, as such models could potentially be used for curation, management, and enhancement of images directly on a cell phone. We saw, that such models can achieve competitive aesthetic ranking performance. And though an aesthetic ranking function is competitive with other aesthetic-based croppers, it is outperformed by methods which focus on capturing image composition quality. Therefore, our last work discussed in Chapter 6 focused on exploring and identifying key aspects of training composition ranking functions using pair-wise ranking optimization.

Chapter 3

Multi-Object Salient Foreground Detection

This work was presented at the Electronic Imaging 2017 conference and can be found at [56]. The full title of the original paper is "Augmenting Salient Foreground Detection using Fiedler Vector for Multi-Object Segmentation". While the following work does not fall under the umbrella of representations or learning approaches for aesthetic inference, this work and algorithms similar to it are often used as a step in computing hand-designed features approximating expert knowledge for aesthetic inference, like the ones seen in the subsequent chapter.

3.1 Introduction

As we move through our daily lives, we are bombarded with an immense amount of visual data. Processing all of this information is physically impossible. However, our brain possesses a mechanism known as visual attention for selecting a subset of the relevant data that we want to focus on. Modeling of visual attention is an extremely important task with many important applications in robotics and computer vision including image compression, object detection, and computer graphics [10].

The notion of relevance in the visual attention models is mainly determined by two processes: bottom-up and top-down processes. Bottom-up attention modeling, also called visual saliency, uses various low-level features including image color, intensity and orientation to determine the contrast of objects with respect to their surroundings [10]. On the contrary, the top-down attention selects the relevant image areas based on task-driven factors such as knowledge, expectation or current goals.

3.2 Related Work

We focus the review on the relevant literature regarding bottom-up visual saliency, especially as it relates to the various saliency estimation approaches that are used to benchmark against the algorithm of [88]. Bottom-up saliency models can in general be described as belonging to one of the following categories: biologically inspired, purely computational and their combination [2]. For a more exhaustive treatment, please see reference [10].

Biologically inspired models, e.g. the model proposed by Itti et al. [38], are often based upon the architecture presented by Koch et al. [50], which used biologically inspired features processed by center-surround operations to determine the saliency score and correctly predict eye fixations.

Computation-oriented models, which use low level image features such as color, emphasize the practical aspect of models such as speed and aim to create saliency maps which segment whole objects and preserve edges [88]. Recently, several models [88, 27, 109, 13, 18, 87] use a variation of super-pixel segmentation methods akin to the SLIC (Simple Linear Iterative Clustering) method [3], to accomplish those goals. Methods such as SLIC over-segment the image into perceptually coherent patches (whose number is much smaller than the number of image pixels) which are able to both preserve the local color information and edges, while abstracting away unnecessary details (i.e. non-significant pixel-to-pixel intensity). Cheng et al. [18] use spatially weighted region contrast to estimate the saliency based on the color histogram differences. Perazzi et al. [87] show the possibility of modeling the saliency estimation in a unified way using high dimensional Gaussian filters, where they combine measures of image patch uniqueness and spatial distribution to estimate the saliency score. Wei et al. [109] build an image graph out of the super-pixel segmentation and estimate the saliency of a patch to be proportional to the shortest path distance from the virtual background node to the said patch. Yang et al. [118] construct an image graph and enforce a background assumption, which assumes most of the borders belong to the background. The authors use a ranking function, which given a query, determines how similar are the remaining nodes to the query nodes. The authors construct a scheme in which they compute the score by determining the saliency of a patch being proportional to the similarity / dissimilarity from the foreground / background queries. Chang et al. [13] use initial saliency maps, measures of objectness, and a measure of how likely an area is to contain an object, to optimize a novel energy function and obtain an improved saliency map.

3.3 Algorithm

3.3.1 Original algorithm

In order to efficiently represent an image, Perazzi et. al [88] use a modified version of the SLIC Superpixel Segmentation algorithm [3] proposed in [87], where the image is segmented into su-

perpixels using k-means clustering in the Color-XY space ([87] uses CIELab color space instead of the traditional RGB space). After Superpixel segmentation, the image is represented as a Graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ also known as the image Region Adjacency Graph (RAG), where each vertex $v \in \mathcal{V}$ is representing a superpixel from SLIC and is assigned a value of the mean Lab color of the superpixel. To model the local relationships in the image, the edge set $\mathcal{E}$ consists of the edges connecting vertices i and j, if their corresponding superpixels share a border in the segmented image. Each edge is assigned a weight that is proportional to the Lab color difference between neighboring superpixels,

$$w_{i,j} = \frac{1}{\|c_i - c_j\|^2 + \epsilon} \tag{3.1}$$

where c_i is a mean Lab color of the i^{th} superpixel and ϵ is a small constant (e.g., $\epsilon = 10^{-4}$) to ensure the numerical stability of the algorithm, in case the color difference is too small. In order to represent the assumption that most of the border pixels belong to the background, Perazzi et al. [88] augment the graph $\mathcal{G}$ with a background node b, which is assigned the mean Lab color of the boundary. A set of edges and their weights that connect the background node and the superpixels on the border of the image are computed by equation 3.1.

In order to assign saliency score to each of the superpixels of the image, Perazzi et al. compute the eigendecomposition of the graph Laplacian matrix L of the image RAG. Then the Fiedler vector, the second smallest eigenvector, is used to compute the saliency scores. Given the Fiedler vector f, the saliency score S is computed as

$$S = -sign(f_b) \cdot f \tag{3.2}$$

and S then scaled to the range $[0, 1]$, where f_b represents the entry of the Fiedler vector corresponding to the background node.

Since one of our proposed approaches considers a high dimensional node embedding, we also propose to compute the saliency scores as

$$S(i) = \|\vec{f}_i - \vec{f}_b\| \tag{3.3}$$

where $S(i)$ is the saliency score for i^{th} superpixel, and $\vec{f}_i$ and $\vec{f}_b$ are the embeddings of the i^{th} and background superpixels, respectively.

3.3.2 Augmenting the background prior

There are images in which the background is often very cluttered. In such case computing the edge weights by considering the average background color will fail to capture the background prior effectively by computing very small edge weights, since the average background color will be sufficiently different from each of the border superpixels and thus resulting in an unsatisfying

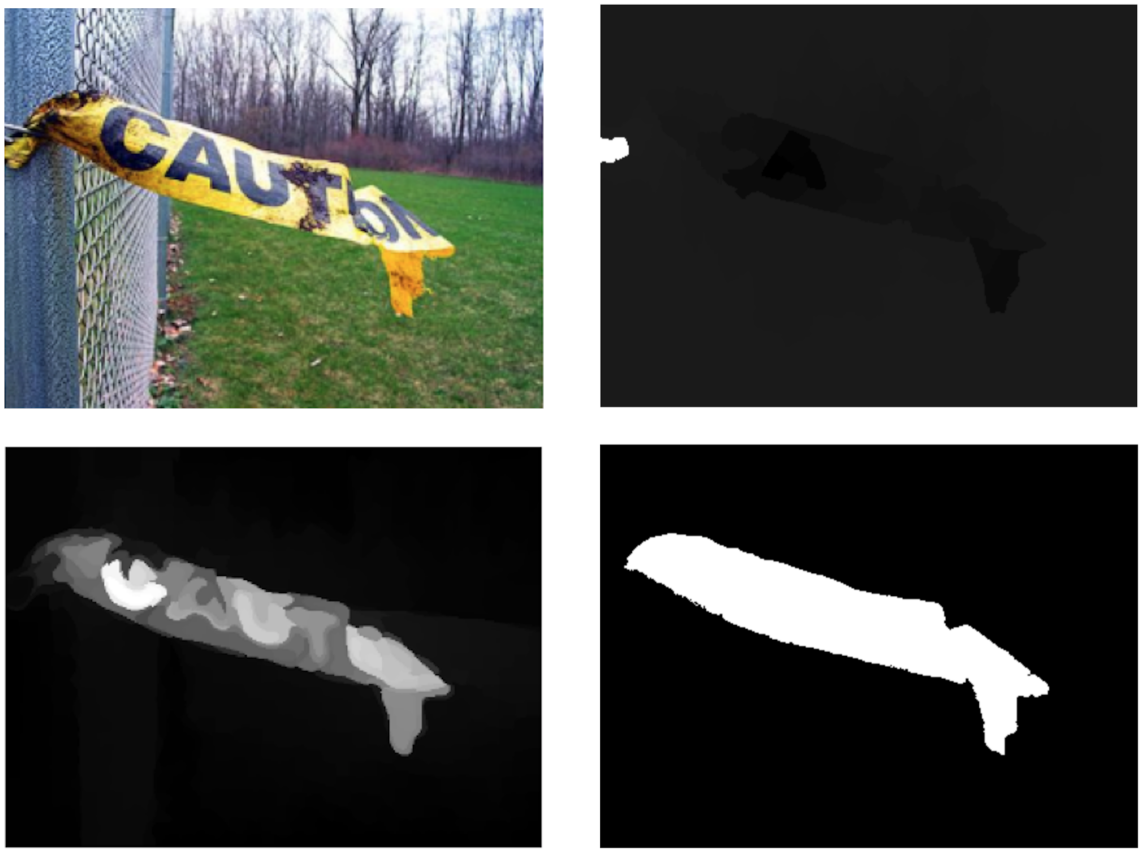


Figure 3.1: Comparison of the saliency maps after augmenting the background prior: original image (top left), Perazzi et al. saliency map (top right), our saliency map (bottom left) and ground truth (bottom right).

saliency map (see the top right image of Figure 3.1). To correct for such a pitfall, instead of assigning to the image background node the average border background color (average color of the border super-pixels), a set of colors representing the background is assigned to the background node. We first perform a K-Means clustering of the border colors and then use the cluster centers, $\{c_1^b, \ldots, c_k^b\}$, to represent the background prior in the node. To compute the edge weight between the background node and the border regions, we simply take the maximum of the weights

computed between region i and each of the k cluster center colors

$$w_{i,b} = max_{j \in \{1,\ldots,k\}} \frac{1}{\|c_i - c_j^b\|^2 + \epsilon}. \tag{3.4}$$

Augmenting the background prior with multiple "colors", we are able to better enforce the background prior as we can see in Figure 3.1 (bottom left).

3.3.3 Detecting multiple objects

To extend the foreground segmentation algorithm to allow for detecting multiple salient subjects in the image, we propose the following schemes: an iterative segmentation scheme and two alternative multi-object foreground segmentation methods which use multiple eigenvectors of the image RAG as an embedding for the nodes and analysis of the presence of additional objects. This embedding is then used to calculate an alternative saliency score. Both of the schemes will use a metric to determine the ideal foreground segmentation. Next, we will describe the Silhouette score and the metric we propose for picking the best saliency map.

Silhouette score

In order to judge the quality of the foreground segmentation, we use k-Means clustering to cluster the saliency scores of each super-pixel into two clusters (Foreground / Background) and then compute a metric knows as the Silhouette score, first introduced by Rousseeuw [91]. The Silhouette score is one possible metric that is used in the interpretation and validation of cluster analysis.

To compute the Silhouette score, we need the resulting clustering and the matrix of distances (or dissimilarities as used by [91]) between the different points (e.g. superpixels and the saliency score assigned to them in our algorithm). For each point i we compute:

- $a(i)$: average distance to the points in the same cluster as i (label that cluster A)
- $D(i, C)$: average distance to the points in cluster C
- $b(i) = \min_{C \neq A} D(i, C)$: by choosing minimum of the $D(i, C)$, we compute the distance to next best cluster assignment for i.

The final score for point i is computed as

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{3.5}$$

which is then combined into a final score f_{sil} for our image by taking the average of $s(i)$ for all of the superpixels.

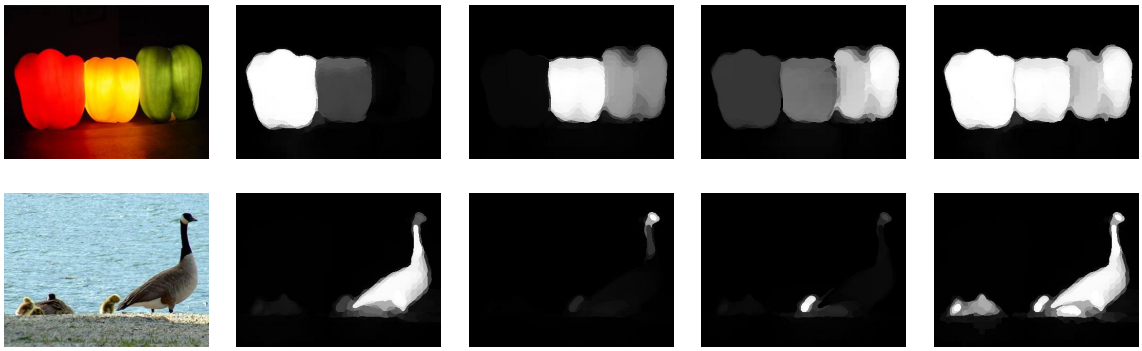


Figure 3.2: Images that show the presence of separate objects / object parts in the higher eigenvector dimensions. From left: Original image, saliency map constructed from first non-zero eigenvector, saliency map constructed from second non-zero eigenvector, saliency map constructed from third non-zero eigenvector, and the final saliency map, whose construction will be described in later section.

Stopping criterion / metric

Both of the multi-object schemes detailed in the next section rely on some sort of stopping criterion / metric, which would determine either the ideal number of iterations or eigenvectors to consider when computing the saliency map for images with multiple objects. In order to determine the ideal iteration / number of eigenvectors, we propose a metric which combines the Silhouette score, f_{sil}, and mean image saliency of the image

$$score_{image} = f_{sil} \cdot \frac{\sum_{x=1}^{m} \sum_{y=1}^{n} S(x,y)}{A(I)} \tag{3.6}$$

where $S(x,y)$ is the image saliency score at the location (x, y) and $A(I)$ represents the area of the image.

Then in order to pick the final saliency map, we choose the map with the highest score defined in equation 3.6.

Presence of objects in eigenvectors

One of the things that we have observed is the presence of multiple salient objects embedded in higher dimensions of the RAG Laplacian matrix eigendecomposition. This can be seen in Figure 3.2, where we show an example of an image and the saliency maps of its eigenvectors (we compute the saliency of an eigevector by computing the scaled distance of each superpixel to the

background node). However the same cannot be said of many of the images that only contain a single salient object, as we can see in Figure 3.3. The Fiedler vector will pick out the most salient object in the image and the subsequent eigenvector (at times several) will contain redundant information regarding the object. Such observations were originally part of the exploration in creating an appropriate stopping metric.

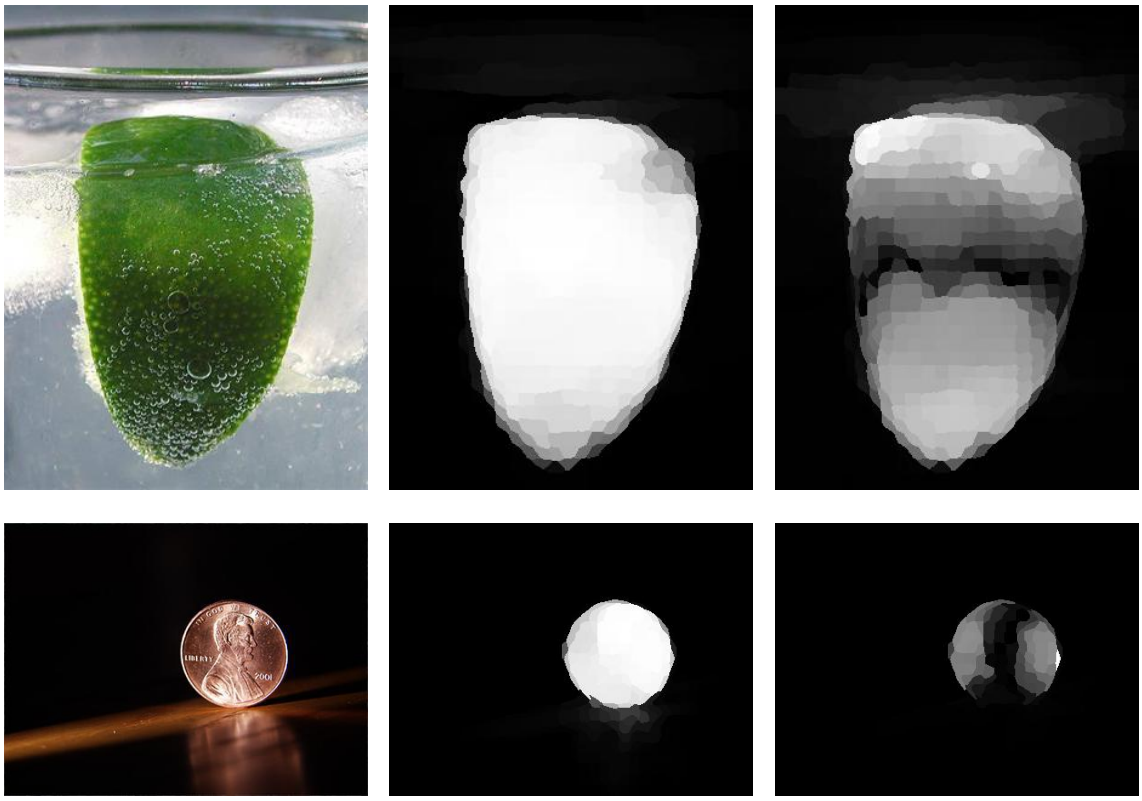


Figure 3.3: Plot of the saliency maps for the first two eigenvectors of the images with a single salient object. From left: original image, first non-zero eigenvector, second non-zero eigenvector.

Stopping criterion based on the eigenvalue difference

A different stopping criterion that we consider is based on the percentage eigenvalue difference between subsequent dimensions. First we compute the full eigendecomposition of the augmented RAG. Then we take a subset of the first k non-zero eigenvalues, and compute the percentage

difference between the subsequent dimensions:

$$\Delta_i = \frac{\lambda_{i+1} - \lambda_i}{\lambda_{i+1}} \tag{3.7}$$

Then in order to get the ideal dimension n, we choose the dimension which produces the largest difference:

$$n = argmax_{1 \geq i < k}\{\Delta_i\}. \tag{3.8}$$

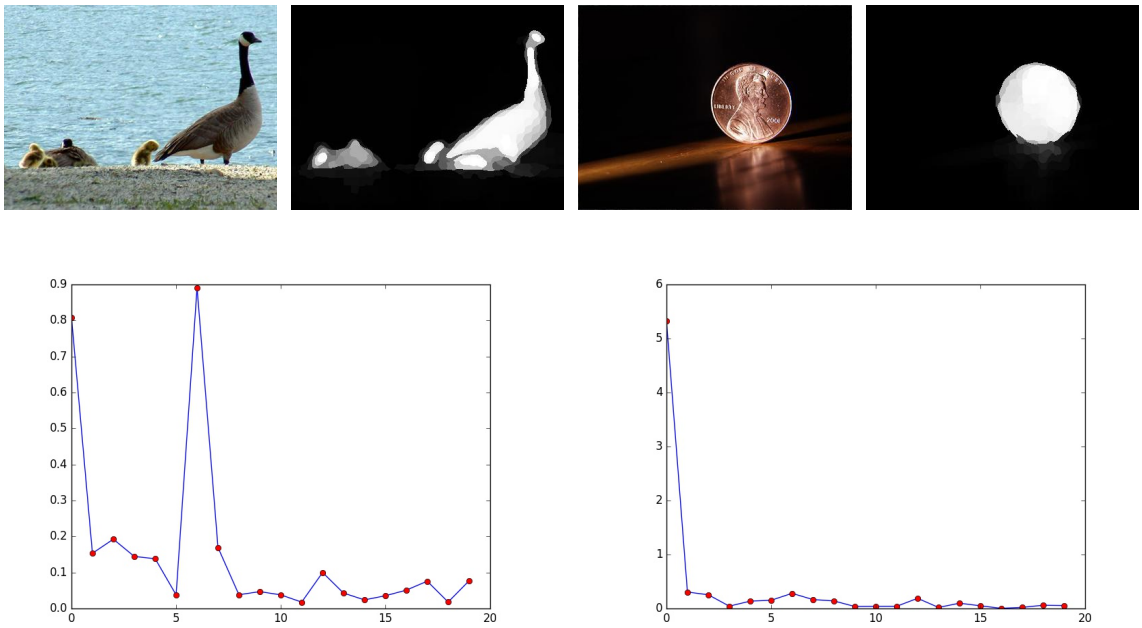


Figure 3.4: Plots showing the eigenvalue percentage difference plots for sample images with single / multiple salient objects.

Multi-object segmentation schemes

The main idea behind the first method, iterative foreground segmentation, is simple: each of the foreground objects are segmented one by one by looking at the most salient object in the image graph at each step of the iteration.

The Iterative Foreground segmentation can be described as:

- Perform an initial foreground segmentation as described in [88] with the improved background prior model, and compute the $score_{image}$ for this map.
- Now, iteratively perform the following steps:
  - Find the set, $\mathcal{S}$, of nodes / super-pixels for which the saliency S_i for super-pixel i is greater than a threshold S_{th}.
  - Modify the image RAG by cutting out the nodes that belong to the set $\mathcal{S}$ (store the saliency scores of these nodes for later processing).
  - Find new saliency scores for the region which remained in RAG by computing the Fiedler Vector of the new graph and computing and modifying it the same way described in [88].
  - Combine the saliency scores of the smaller region with the scores for the nodes from the set $\mathcal{S}$, to obtain the new saliency image and compute its $score_{image}$.
  - Repeat for predetermined number of iterations.
- Choose the map with highest $score_{image}$.

Based on the previous observations of the presence of additional salient objects in different eigenvectors, we prepose two alternative ways of constructing an image saliency map based on considering multiple eigenvectors.

The first method for foreground segmentation proceeds as follows:

- Construct the RAG of the image as described in [88] and augmented with the improved background node.
- Construct the Laplacian matrix of the image RAG.
- Consider the k smallest eigenvectors corresponding to non-zero eigenvalues and use them as a k-dimensional embedding of the graph nodes.
- Calculate the new saliency scores by:
  - Calculating the distance between the k-dimensional embedding of the background node and node i.
  - Renormalize all of the distances to lie in the range between $[0, 1]$, which will give us the relevant saliency scores S.

- Compute a metric for maps created by considering projections with varying number of eigenvectors (we consider up to four eigenvectors for the embedding of our graph) and choose the map with highest score achieved by the metric.

In order to observe the map chosen by the score defined above, please refer to the Figure 3.5 and Figure 3.6, which show examples of the original images and the corresponding sequences of saliency maps.

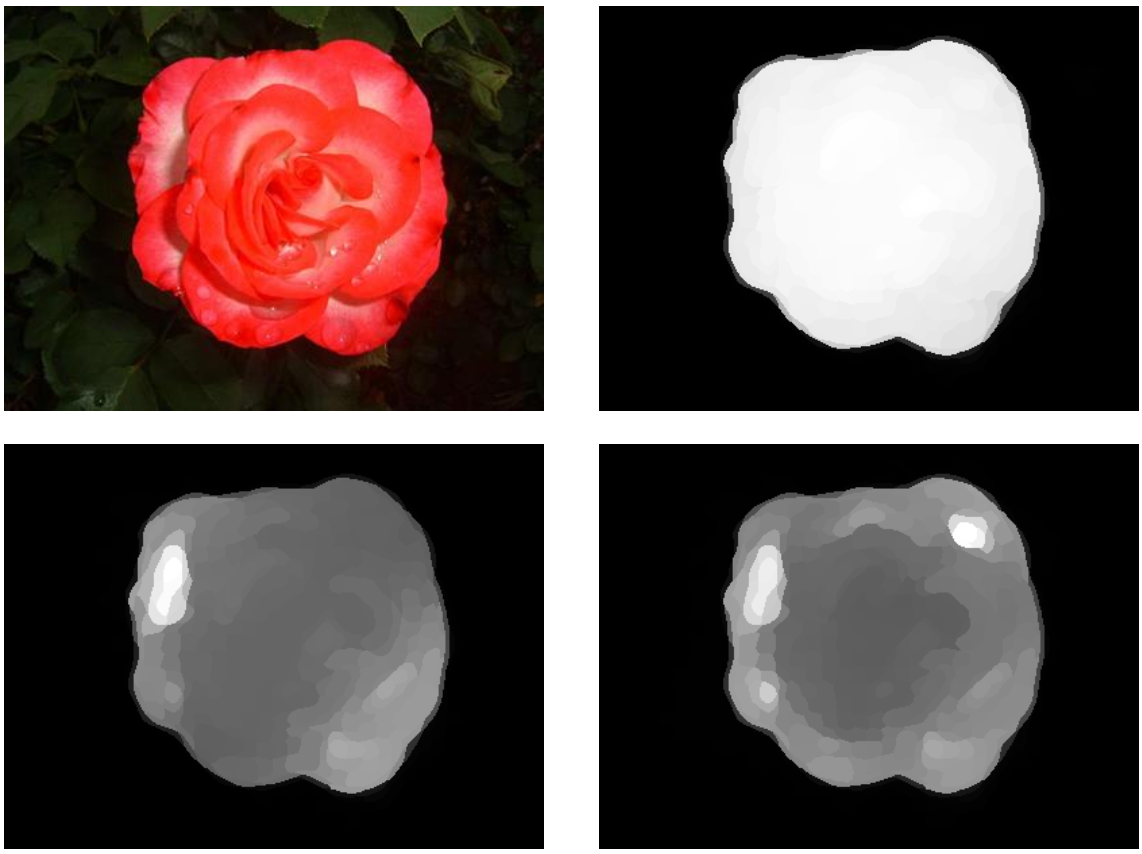


Figure 3.5: Original image (top left) of a scene with one salient object and its corresponding saliency maps as we vary the number of eigenvectors considered for the superpixel embedding: 1 (top right), 2 (bottom left), 3 (bottom right). Map with 1 eigenvectors was chosen as the best by our score.

For the purpose of binarizing a floating point image, we will utilize the adaptive threshold

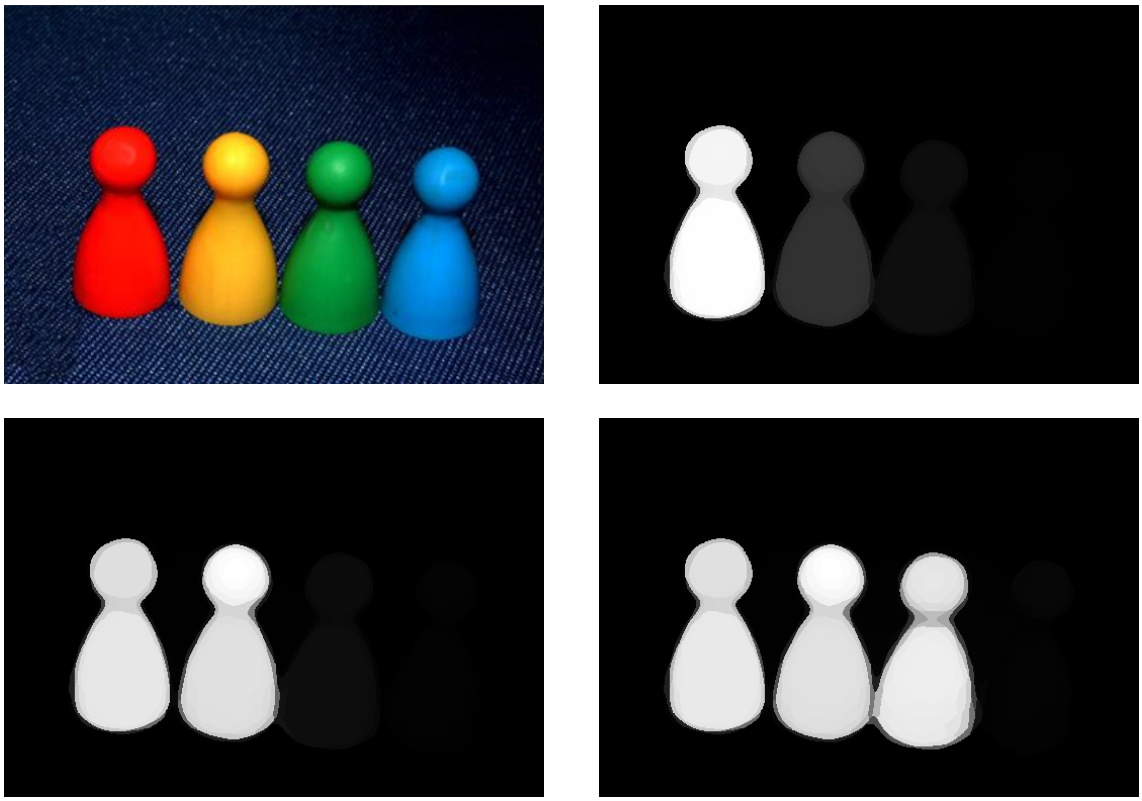


Figure 3.6: Original image (top left) of a scene with multiple salient objects and its corresponding saliency maps as we vary the number of eigenvectors considered for the superpixel embedding: 1 (top right), 2 (bottom left), 3 (bottom right). Map with 3 eigenvectors was chosen as the best by our score.

proposed in [2] defined as twice the mean image saliency:

$$T_a = \frac{2}{W \times H} \sum_{x=1}^{m} \sum_{y=1}^{n} S(x, y) \tag{3.9}$$

Secondly, the following method first computes the desired number of eigenvectors to consider and subsequently constructing the saliency map in the following way:

- First precompute the number, n, of eigenvectors to consider using equation 3.8.

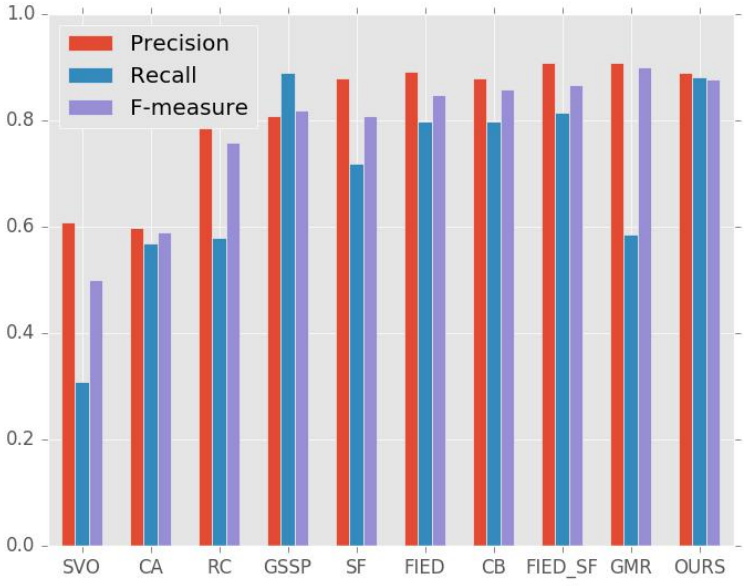


Figure 3.7: Benchmarks. Performance of the various algorithms on the MSRA [2] dataset.

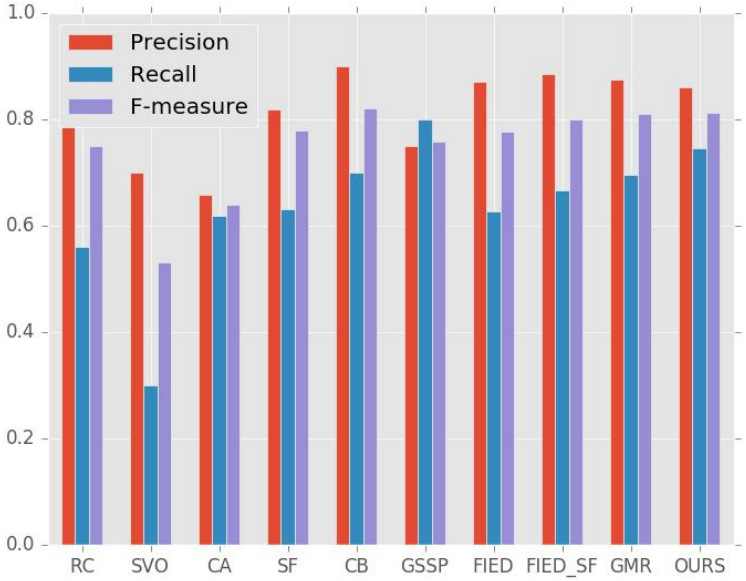


Figure 3.8: Benchmarks. Performance of the various algorithms on the ImgSal [63] dataset.

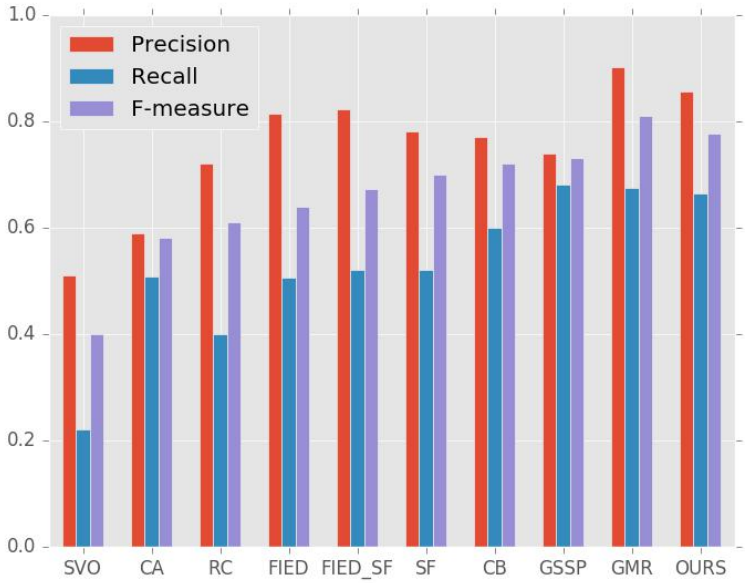


Figure 3.9: Benchmarks. Performance of the various algorithms on the SED1 [4] dataset.

- Compute the vector of saliency scores, S, for the superpixels using the improved background prior.
- If the n = 1, then we are done otherwise repeat the following procedure for $n \geq 2$. Assume we have computed the saliency scores for the first $k, k < n$ dimensions, which we will call S_k. To incorporate the $k + 1^{th}$ dimension in the computation of the final saliency scores S, proceed as follows:
  - Compute the saliency scores for the $k + 1^{th}$ dimension, S_{k+1} by computing the distance of each superpixel to the background node and rescaling the score between $[0, 1]$.
  - Compute the threshold T_a^{k+1} based on S_{k+1} and extract the set of superpixels i for which it is true that $S_{k+1}^i \geq T_a^{k+1}$ and call the set $\mathcal{N}$.
  - For $i \in \mathcal{N}$, let $S_{k+1}^i := \max\{S_k^i, S_{k+1}^i\}$, otherwise $S_{k+1}^i := S_k^i$.
  - If $k + 1 < n$, repeat the procedure, else construct the image saliency map.

3.4 Results

In order to provide a direct comparison of our algorithm with the original version proposed in [88], we evaluate the algorithm on the same three datasets used in the original paper: MSRA [2], SED1 [4] and ImgSal [63].

In order to benchmark the results of our algorithm, we will compare to the results obtained by Perazzi et al. (FIED) [88] and reporting the results published in [88] for the recent top-performing methods that include: context-aware saliency (CA) [27], context-prior (CB) [37], geodesic saliency (GSSP) [109], generic objectness (SVO) [13], global-contrast (GC)[18], graph manifold ranking (GMR) [118], and saliency filters (SF) [87] and their combination with FIED (FIED_SF).

3.4.1 Quantitative results and evaluation

To compare our algorithm with the above mentioned algorithms, we create binary maps from the computed saliency maps by first computing the adaptive threshold T_a of equation 3.9 proposed in [2] and assigning the values above and below T_a to the foreground and background classes respectively. We evaluate the proposed algorithm by computing the Precision, Recall and F-measure of the binary saliency maps compared to the ground truth maps. The F-measure is computed by

$$F_\beta = \frac{(1+\beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{3.10}$$

where $\beta^2 = 0.3$ to emphasize the importance of precision as seen in previous experimental setups [2, 88, 118].

The performance evaluation on the three datasets is shown in Figures 3.7, 3.8, and 3.9, where we benchmark our algorithm with the augmented background model combined with the last multi-object extraction method (as it is the best performing foreground extraction method). As we can see from Figures 3.7, 3.8, and 3.9, we achieve comparable results to the original algorithm for the MSRA and ImgSal datasets and a slight improvement for the SED1 dataset in terms of precision, which is defined as the fraction of retrieved pixels that actually belong to the foreground. Further, we see a good improvement in the recall value, which can be attributed to the improvement in extraction of multiple subjects, as recall is defined as the ratio of correctly detected pixels compared to the ground truth.

3.5 Limitations

Although we were able to augment the algorithm, the new algorithm still has difficulty with detecting foreground objects whose color is too similar to its surroundings. Furthermore, the first two

foreground extraction methods rely on the image metric to pick the best saliency map. A problem arises when taking the next step results in a larger increase in the average saliency than the decrease in the quality of the map (Silhouette score). In such a case, the algorithm might choose the worse map, and thus one of the possible avenues for future work is to explore alternative stopping criteria.

3.6 Conclusion

We proposed several improvements to a graph-based foreground detection method. First, we showed that by modeling the background to consist of several colors can lead to an improved foreground extraction. Furthermore, we have presented three approaches and shown their ability in segmenting multiple salient objects. The evaluation of the algorithm showed equivalent / slightly improved results in precision and improvement in the recall over the original algorithm as can be seen from the benchmarking results.

As part of the future work we would like to gain a more thorough understanding of the spectral properties of the image graphs. Furthermore, we would like to explore several methods to enhance graph creation process, in which we could incorporate different shape priors to alternate the edge creation process. Several deep learning methods were recently developed which allow for processing of graphs, knows as Graph Convolutional Neural Networks [49]. We would like to further explore the application of such methods to foreground detection using a reduced image representation with the Region Adjacency Graph.

Chapter 4

Expert knowledge for image aesthetics

The following work was published in the Transactions on Image Processing under the following full title "Leveraging Expert Feature Knowledge for Predicting Image Aesthetics" [57]. The previous chapter directly feeds into the following work, as saliency estimation was used for approximating some of the expert rules. One of the main goals of this work was to bridge the gap between traditional and deep learning approaches, and explore the possibility of improving the predictive performance of deep learning features by fusion with hand-designed features. We first explore the landscape of hand-designed features from previous literature. We show that these features can rival predictive performance of deep learning features. Lastly, we show that fusing hand-designed features can indeed provide an improvement in predictive performance over baseline deep learning features.

4.1 Introduction

The problem of determining the aesthetic appeal of an image is challenging because the overall aesthetic value of an image is dependent on its technical quality, composition, emotional value, etc. In determining the aesthetic value of an image, the algorithms follow a similar pipeline to other branches of computer vision, such as object detection: a set of image features is extracted from an image which is then used as an input to a classifier or regressor for further processing.

Many of the early algorithms for inference of image aesthetics relied on carefully chosen and crafted features based on expert knowledge, e.g. established photographic rules [5, 121] such as those seen in Figure 5.1. These, however, went out of favor and were replaced by generic features based on various local descriptors and convolutional neural networks. Although these networks are superior in their capacity to learn high-level semantic information from low-level pixel information, it is possible that the networks may not discover some essential knowledge, e.g. global

(a) Rule of Thirds [122] (b) Symmetry [121]

Figure 4.1: Common photographic rules used in capturing aesthetically pleasing photographs.

texture information contained in the gray-level co-occurrence matrix, even when appropriate optimization is in place.

In this chapter we conduct a comprehensive study of hand-designed features that rely on expert knowledge from various fields, and we explore the extent to which hand-crafted features aid learning-based features in predicting image aesthetics. The major contributions of this investigation are listed as follows:

- We analyze and compare a wide variety of hand-crafted features in their ability of predicting continuous and binary aesthetic scores.
- We perform feature elimination for various tasks (classification, regression, categories) to uncover the best performing features for them.
- We investigate the possibility of fusing hand-crafted features with learned features for improving aesthetic inference.

The remainder of the chapter is organized as follows. In Section 2. we summarize the datasets used in this work. Section 3. presents the analysis of the ability of hand-designed features to predict aesthetics and performs feature elimination to surface the best performing features. Section 4. explores the possibility of fusing hand-crafted features with learned features from convolutional neural networks. Concluding remarks and suggested future work are discussed in Section 5.

4.2 Datasets

In the process of computing features and evaluating the algorithms, the following datasets are used: Aesthetic Visual Analysis (AVA) [80], CUHKPQ [100], HiddenBeauty [94], and a Kodak Aesthetics dataset [41].

The *CUHKPQ* dataset contains more than 17,690 images divided into seven semantic categories with binary labels indicating high or low quality. In order to assign labels to the images, each image was viewed by ten people who labeled the image as high or low quality. An image was kept and assigned a final label if at least 8 out of 10 people agreed with their assessment of the image. We primarily use the CUHKPQ in the computation of image features that required reference high/low quality data. For example, Ke et al. [47], one of the algorithms considered here, computes a color distribution feature, which calculates the number of high-quality photos retrieved by the nearest neighbor search.

The *Hidden Beauty of Flickr Pictures* (HiddenBeauty) dataset was collected as part of an effort to surface the "hidden gems" among the pictures that have very low popularity / interestingness as measured on Flickr [94]. More than 15,000 images were chosen from the sample of nine million images from the larger YFCC100M dataset [103]. Although the HiddenBeauty database is not the largest database, we chose to use it because each image was rated on a five-point scale based on metrics that were clearly described to each rater, thus minimizing the bias of what is considered "high quality". The labels were collected via the CrowdFlower crowdsourcing platform. Each image was labeled by at least five different people, with each evaluator having a top track record on the platform. Each image belongs to one of four categories - human, nature, urban, people - and its aesthetic score is the mean rating of all of the scores.

The Kodak Aesthetics dataset is an extended version of the dataset described in Jiang et al. [41]. The dataset was created to resemble the variety of images found in consumer photography. It consists of more than 1,500 images each rated by four people on the 1-100 scale. The ground truth score for an image is the average of its four ratings.

The *Aesthetic Visual Analysis* (AVA) dataset is one of the largest datasets available for working with aesthetic preferences [80]. The images were sourced from *www.dpchallenge.com*, a website housing a community of amateur and professional photographers. Each image in the dataset received between 78 and 549 votes per images, with an average of 210. Each image is given a score on a scale of $1 - 10$. The average rating is considered to be the ground truth aesthetic score for the image. Along with aesthetic ratings, images come with 66 semantic and 14 photographic style annotations (e.g. High Dynamic Range, Soft Focus, etc.).

Types of Features	Feature	Description
High-Level Features	Face Detection [100]	Using a detector to uncover the presence of faces in the image
	Face Shadow [100]	Approximating the quality of lighting on faces.
	Average Region Saliency [94]	Measures salience of objects in different parts of the image
Affect	Affective Dimensions: Pleasure, Arousal, Dominance	Indicators of emotion calculated by combining average Saturation and Brightness as defined by [74]
	HSV Statistics [20, 94]	Measure of the mean and spread of the HSV image channels.
Aesthetics	Rule of Thirds [20, 100, 94]	Guideline in photography for placing the subject within the image.
	Depth of Field [20]	How well is the background separated from the foreground?
	Colorfulness [20]	How different is the color distribution in an image from an ideal one.
	Color Harmony [8]	Features approximating how pleasing are different color combinations.
Texture	GLCM Entropy & Skewness	Measures of texture based on Gray Level Co-occurrence Matrix
	Wavelet-based Texture [20]	Measures of spatial smoothness based on Daubechies wavelets

Table 4.1: Type, name and description of the variety of features that the algorithm considers

4.3 Aesthetic Assessment with Hand-crafted features

To better understand the utility of hand-crafted features in aesthetic assessment, this section compares the performance of a selection of algorithms and investigates an approach for selecting a subset of features. For our investigation, we selected algorithms enabling a wide variety of image features to be considered, e.g. different measures of photo quality such as image blur [20, 47, 65], image composition and content [72, 100] and generic features [77, 65]. Table 4.1 describes a selection of features considered. For specific details on the individual features in each feature set, see the selected references. The features selected for comparison originate from the following feature sets:

1. Datta et al. (DATTA)[20]

2. Ke et al. (KE) [47]
3. Subject-based Photo Quality (PQ) [72]
4. Content-based Photo Quality (CBPQ) [100]
5. Aesthetics using Generic Image features (FV) [77]
6. Yahoo Complete Framework for Image Aesthetics (YCF) [65]
7. EPFL Context Image Aesthetics (Global features) (EPFL) [97]
8. Video Aesthetics (VQ) [8]
9. Yahoo HiddenBeauty Algorithm (YHB) [94]

Both pre-trained CNN features and the various generic features (e.g. SIFT) are either used independently to predict the quality of images or to create higher-level meta-features. In total, we extract a total of 331 numerical features: 54 for DATTA, 10 for KE, 10 for PQ, 16 for CBPQ (20 for images with humans), 27 for YCF, 14 for EPFL, 149 for VQ, and 47 for YHB. Additionally, we extract the Fisher Vector descriptors as described in [77] and introduced in [89]. In implementing the DATTA algorithm, we avoid the features labeled $f8$ and $f9$ in the paper because the computation of image uniqueness (computed as the mean distance to the top 20 and 100 closest matches) was tied to a selection of 1000 images which were unavailable. Additionally, for the computation of the human-related features in CBPQ, we use the deep learning based face detector in the *dlib* machine learning library [48] which is more accurate in terms of detection performance and false positive retrieval.

4.3.1 Learning framework

Often one of the most important aspects of building predictive systems is the selection of a suitable learning framework that achieves good generalization performance and speed of evaluation. Earlier algorithms use a variety of techniques to predict the aesthetic image descriptor, e.g SVM, Neural Networks or Random Forest. To understand the differences and performance of individual algorithms and features, the same learning framework is used on top of each feature set. The model / learner in use is an ensemble learning technique known as Gradient Boosted Trees as it achieves excellent generalization performance supported by both theory and practice [14]. An advantage of using a Boosted Tree learner is its ability to simultaneously quantify the importance of features and train a model. The notion of feature importance is measured by three metrics: gain, cover, and frequency [26].

4.3.2 Methodology

To evaluate the performance of the chosen algorithms and individual features, a separate model is trained for each algorithm. In order to maximize the utility of the datasets, we evaluate the

performance of the algorithms using k-Fold Cross-validation (CV). In k-Fold CV, the dataset $\mathcal{D}$ is split into k non-overlapping folds / parts $\mathcal{D}_1, \ldots, \mathcal{D}_k$ (we chose $k = 10$ based on [51]). To estimate the performance metric on a dataset, the learner (or inducer $\mathcal{I}$ as in [51]) is trained and tested k times. Given a fold $i \in \{1, \ldots, k\}$, the model is then trained on the dataset $\mathcal{D} \backslash \mathcal{D}_i$ and tested on $\mathcal{D}_i$. Thus we iterate through each fold, treating it as the test set and the other $k-1$ folds as the training set for our learner, allowing us to utilize each image in both training and testing. The cross-validation estimate of accuracy is given by calculating the accuracy / correlation coefficient for the concatenated vectors from all folds [51] as

$$acc_{CV} = \frac{1}{m} \sum_{\langle x_i, y_i \rangle} \delta \left(\mathcal{I} \left(\mathcal{D} \backslash \mathcal{D}_i, x_i \right), y_i \right), \tag{4.1}$$

where m is the total number of samples in the dataset, and $\mathcal{I} \left(\mathcal{D} \backslash \mathcal{D}_i, x_i \right)$ is $\hat{y}_i$, the label given to x_i that was trained on the dataset consisting of all of the folds except one that includes x_i. To prevent bias towards any algorithm or feature set, the training/prediction of the aesthetic scores was performed across all tests with the same parameters[1] (other learner parameters are set at their default values). In training the models to predict the continuous aesthetic score, we optimize the mean square error

$$l_\theta = \sum_i \left(y_i - \hat{y}_i \right)^2, \tag{4.2}$$

and for predicting the binary High / Low score we optimize the logistic loss

$$l_\theta = \sum_i \left[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \right], \tag{4.3}$$

where y_i is the ground truth aesthetics score, $\hat{y}_i$ is the predicted aesthetics score, and θ are parameters optimized to achieve the best performance for a given loss function. We quantify the performance of each trained model by either calculating the average accuracy for binary labels as defined in Equation 4.1 or calculating the predicted aesthetic score for each fold and then calculating the correlation coefficient between the predicted and ground truth values for each image.

4.3.3 Comparing different algorithms

First, we compare the different algorithms to each other in terms of their ability to predict the continuous aesthetics scores of the HiddenBeauty and Kodak datasets. The YCF algorithm contains a feature which predicts the probability of an image being high quality based on the deep learning features extracted from the second-to-last layer of a CNN (ImageNet pre-trained VGG16 model

[1] The particular values for the XGBoost Tree Learner [14] are as follows: max_depth, n_estimators, subsample, colsample_bytree, and colsample_bylevel are respectively set at 8, 100, 0.9, 0.9, and 0.9

	DATTA	KE	PQ	CBPQ	FV	YCF-HC	YCF	EPFL	VQ	YHB	NoDLFV	All	CNN
HB	0.413	0.431	0.296	0.458	0.258	0.478	0.540	0.421	0.434	0.453	0.568	0.600	0.589
Kodak	0.571	0.547	0.293	0.297	0.310	0.589	0.67	0.384	0.509	0.548	N/A	0.733	0.636

Table 4.2: Comparison of the algorithm performance in predicting aesthetics score in terms of the correlation coefficients for the HiddenBeauty and Kodak datasets.

[98]). Therefore, in addition to considering the nine algorithms detailed above, we train three additional models: one which only considers the scores predicted by a model trained on CNN features, one with only the hand-crafted features from YCF, and a model which considers all features except those that predict the probability of being high quality image based on the CNN features, and SIFT and Color Fisher Vectors (denoted "NoDLFV" in Table 4.2). Table 4.2 shows the model performance for the various feature sets as evaluated by the correlation coefficient for each model.

As we can see from Table 4.2, even features crafted by early algorithms are effective for predicting photo quality, as evidenced by the competitive correlation coefficient we see for the first two algorithms in 2006 (DATTA, KE) as compared to the more recent methods (VQ, YHB). If we compare the performance of the algorithms between the two datasets, we see that although the algorithms perform better on the Kodak dataset, the algorithms generally exhibit the same trend in the performance on both datasets. The better performance of algorithms on the Kodak dataset can be explained by the way scores were obtained: in the Kodak dataset, each image was scored by the same four people, as opposed to the CrowdFlower platform where a diverse group of people rate each image (resulting in a larger variance in the scoring from image to image).

The last column of Table 4.2 shows the results for the pre-trained CNN features, obtained as described above from the ImageNet pre-trained VGG16 model. We can see that, despite the absence of fine-tuning, the features perform very well, giving us the second/third best results among all of the feature sets for the HiddenBeauty / Kodak dataset respectively.

One of our original goals was to observe ways of combining hand-crafted features and deep learning, and thus we observe two models: "NoDLFV" and "ALL", which will be described in the next section. Combining all hand-crafted features results in an improvement as compared to the YHB (the best single algorithm). Furthermore, one of the ways we can combine HC features with DL, is to use pre-trained CNN features to compute a quality meta-feature, and then concatenate it to them, resulting in improvement in performance of predicting aesthetics as can be seen in Table 4.2.

4.3.4 Feature Elimination

Although combining all of the features improves results by a small margin, computing the features for all algorithms is computationally inefficient. Therefore, in this section we investigate how

many of the features are actually needed to achieve a good performance in predicting aesthetics on the HB dataset. To determine the top features, we will perform *Recursive Feature Elimination* (RFE) [30], which itself is an instance of Backward Feature Elimination [52]. RFE is an iterative procedure and can be simply described as the following sequence of steps [30]:

1. Train the classifier (optimizing the parameters θ with respect to a loss function l_θ)
2. Compute the ranking criterion $\mathcal{D}(X_i)$ for each feature X_i
3. Remove the feature with the smallest ranking criterion.

At each step, we train a learner to predict the aesthetics scores and, based on the ranking of all features, we remove the feature with the lowest ranking criterion and retrain the model with remaining features. We use the gain of each feature (defined as "improvement in accuracy brought by a feature to the branches it is on"[26]) as a ranking criterion for the individual features as it is the most intuitive way to measure feature importance among the three metrics. The gain-based ranking criterion is similar to the Mean Decrease Impurity importance (MDI) [67]

$$\mathcal{D}(X_m) = \frac{1}{N_T} \sum_{T} \sum_{t \in T: v(s-t)=X_m} p(t) \Delta i(s_t, t), \tag{4.4}$$

where $i(t)$ is any impurity measure. In our case the impurity or gain of each feature will depend of the loss function used to optimize the learner.

In order to observe how the performance of the model changes with each removed feature, at each step we perform a 10-fold cross-validation on the current features where we predict the mean aesthetic score. For a more formal description of RFE, please see Section 3.2 of [30].

Aesthetic inference

Figure 4.2 shows the variation of r^2 in predicting the mean aesthetic score with respect to the number of considered features (the abscissa covers a shorter range, since there is no change in performance for more than 100 features). As we can see in Figure 4.2, many of the features contain information that could be considered complimentary and, thus, do not improve the performance past roughly 40 features. As can be expected, "ALL" features perform slightly better in terms of predicting the overall score than the "NoDLFV". Although the performance of the "NoDLFV" features is arguably constant even with 250 features removed, we see a much sharper drop in the performance as compared to "ALL" if we keep removing additional features past this point. This could indicate the strength of the DL Probability feature and how much the "quality" of the image is related to the aesthetics score.

Figure 4.2 shows "Some8" and "No8" feature sets in which part or all of the features from VQ algorithm were respectively removed in order to study its impact on the regression performance

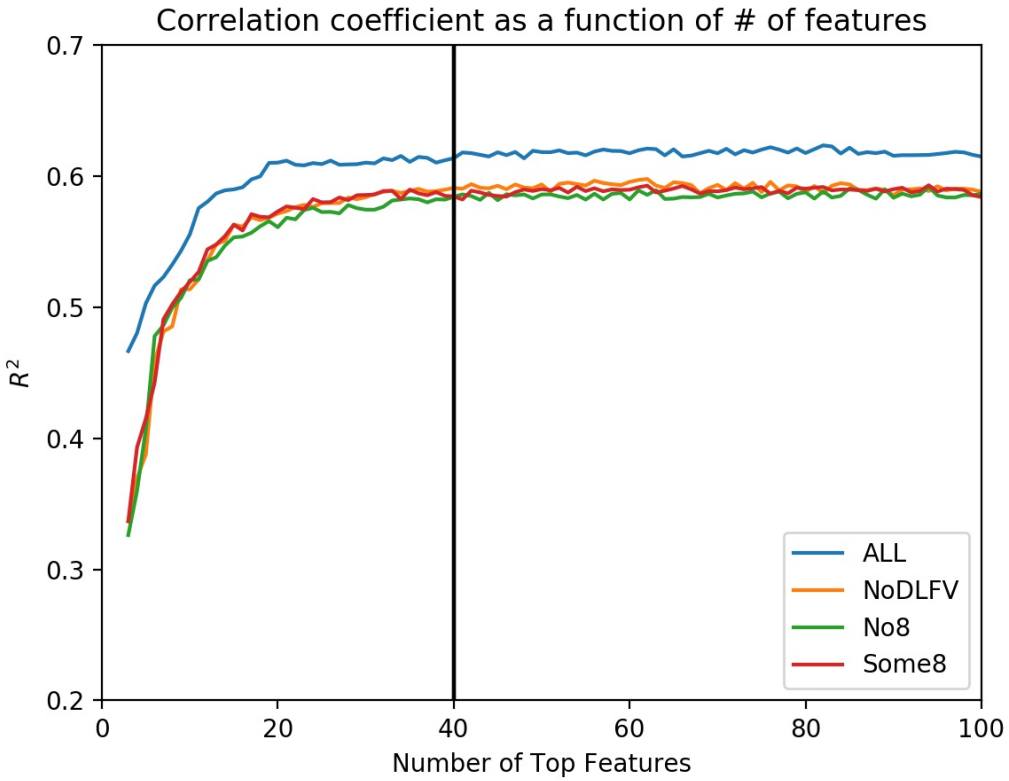


Figure 4.2: Regression performance as the function of top k features on the HB dataset. The vertical line at $k = 75$ indicated a point after which the regression performance remained approximately constant.

since when observing the top 75 features, a large proportion of the features came from this algorithm. As can be seen, the performance of the learner is the same and thus corroborating the notion of the complementarity/redundancy of some of the features. This can further be seen by comparing the top features of the different feature sets. In NoDLFV, many of the features from VQ pertained to Color Harmony and Colorfulness. Once all of the features from the VQ algorithm were removed, features describing similar information took their place, e.g. the Color Harmony and Hue Complexity features from CBPQ [100].

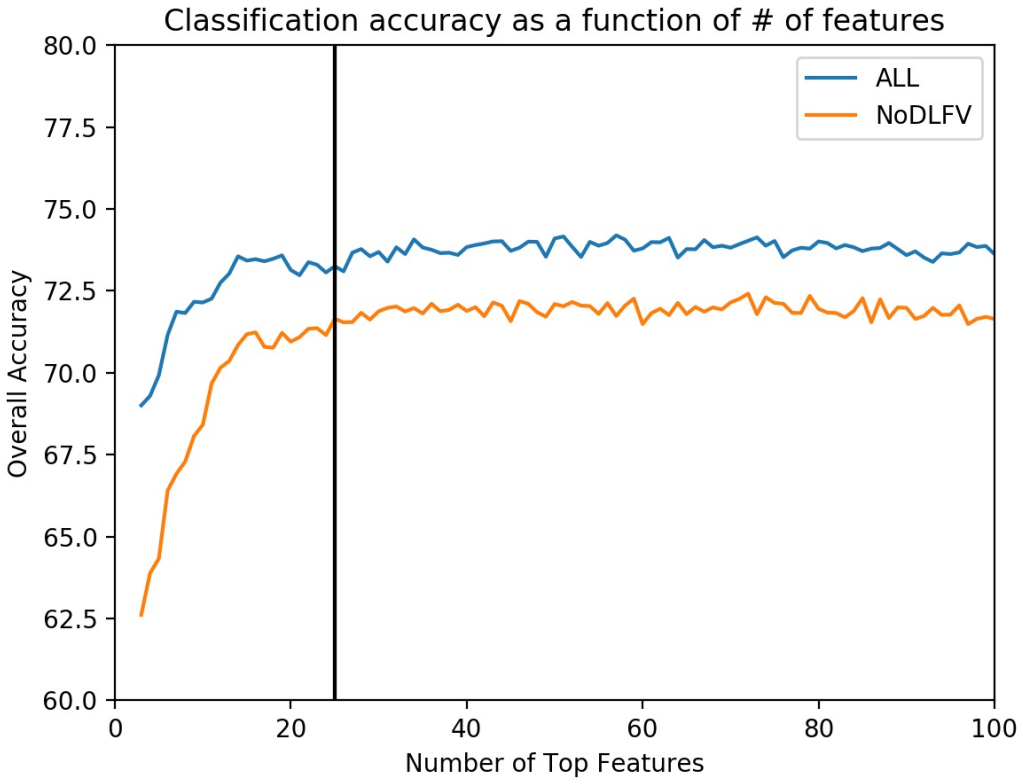


Figure 4.3: Classification performance as the function of the number of top k features HB dataset. The vertical line at $k = 25$ indicated a point after which the classification performance remained approximately constant

Binary Classification

In order to perform classification, the HiddenBeauty dataset is split at the mean score μ, i.e. images with scores $\geq \mu + \delta$ are assigned to the "high" quality class and images with scores $\leq \mu - \delta$ are assigned to the "low" quality class. Then we compute the binary accuracy, as defined in (4.1), of the predicted labels for the HiddenBeauty dataset based on the 10-fold CV, as described previously in section 4.3.2. Figure 4.3 shows the classification performance as a function of the top k features. Similar to regression, the "ALL" features achieve a better performance, due to the DL probability feature as can be seen in Table 4.3. It was found that, for classification, only 25 features are needed to achieve the full performance as opposed to the top 40 for regression. This can be attributed to

Regression		Classification	
ALL	NoDLFV	ALL	NoDLFV
A6 f25: DL Probability	A2 f5: Blur	A6 f25: DL Probability	A2 f5: Blur
A2 f5: Blur	A8 f92: White Balance 6	A2 f5: Blur	A8 f92: White Balance 6
A6 f26: Sift FV Probability	A1 f55: V DOF Ind	A1 f54: S DOF Ind	A9 f17: Itten H07
A2 f9: EdgeDist_Low	A4 f15: Spatial Complexity Rela	A9 f17: Itten H07	A9 f27: Itten S05
A8 f92: White Balance 6	A2 f9: EdgeDist_Low	A6 f26: Sift FV Probability	A2 f9: EdgeDist_Low
A4 f9: Dark Channel	A1 f22: Size feat	A2 f7: Tong_BlurExtent	A2 f1: BBox_Edges
A1 f1: mean intensity	A2 f1: BBox_Edges	A2 f9: EdgeDist_Low	A1 f22: Size feat
A4 f15: Spatial Complexity Rela	A2 f7: Tong_BlurExtent	A9 f13: Itten H03	A9 f14: Itten H04
A1 f5: ROT mean H	A4 f9: Dark Channel	A8 f92: White Balance 6	A4 f14: Spatial Complexity Bkgd
A2 f7: Tong_BlurExtent	A1 f7: ROT mean V	A4 f9: Dark Channel	A1 f55: V DOF Ind

Table 4.3: Top 10 performing features for regression / classification on ALL / DLFV features sets.

the complexity of predicting the continuous aesthetic scores: trying to predict the aesthetic scores is much harder task, since notion of beauty or aesthetics of an image is very subjective and thus we are likely to be learning some of the underlying noise. Dividing the images into high or low quality and predicting binary label is an easier task. Table 4.3 lists the top ten features (as measured by the gain factor described above) for the different tasks and feature sets tested for classification and regression. Interestingly, the top two performing features do not change from regression to binary classification. In all tasks and datasets, features that measure technical quality of images (Blur or White Balance) rank very high in importance. However, in classification, we see much higher importance placed on features related to color (Itten).

4.3.5 Model and feature analysis by categories

All	Animals	Nature	People	Urban
0.665	0.678	0.508	0.592	0.617

Table 4.4: Comparison of the hand-crafted feature performance in predicting aesthetics score in terms of the correlation coefficients for the HiddenBeauty image categories.

The content of images is known to affect the attributes that describe the image the best [100]. In our analyses of the category-specific features, we consider the HiddenBeauty dataset since it separates all of its images into one of following categories based on content: people, urban, nature, and animals. For the purpose of this investigation, we assume that category of an image is known to us beforehand. We measure the importance of all the features by first concatenating the feature extractions from the explored feature sets and then use previously described RFE to uncover the top performing features for each category.

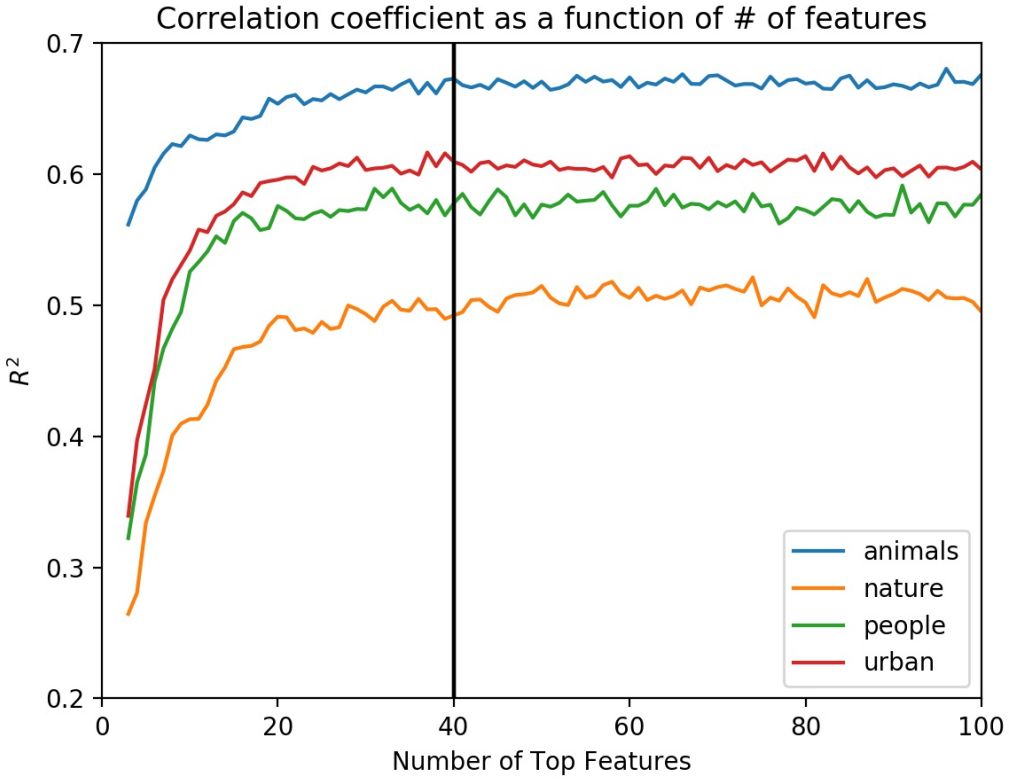


Figure 4.4: Regression performance as the function of the number of top k features the different categories of the HB dataset. The vertical line at $k = 40$ indicated a point after which the regression performance remained approximately constant

The resulting r^2 values computed with 10-fold CV for each category are listed in Table 4.4. As we can see, images of animals achieve the best performance in predicting their aesthetic score, followed by the urban, people and nature categories. The nature category proves to be particularly difficult due to the diversity of its contents: the nature category includes a wide variety of images depicting landscapes, plants and animals (i.e. smaller animals such as bees on a flower or a bird flying around tree). Figure 4.4 shows the regression performance as a function of the number of features. Similar to regression with all of the categories, the performance stays roughly the same, until the number of remaining featuring is ~ 40, after which we see a drop-off in the performance with decreasing number of features. It is interesting to observe the drop-offs, because as the

Animals	Nature	People	Urban
A6 f25: DL Probability	A6 f25: DL Probability	A6 f25: DL Probability	A6 f25: DL Probability
A2 f1: BBox_Edges	A2 f5: Blur	A2 f5: Blur	A9 f30: Itten V03
A2 f9: EdgeDist_Low	A9 f27: Itten S05	A2 f9: EdgeDist_Low	A4 f16: Hue Complexity
A2 f5: Blur	A9 f34: Symmetry	A2 f7: Tong_BlurExtent	A8 f1: Sal Region Area
A4 f9: Dark Channel	A1 f7: ROT mean V	A8 f92: White Balance 6	A2 f5: Blur
A8 f10: Dark Channel 5	A1 f4: mean hue	A1 f6: ROT mean S	A6 f21: Noisiness
A7 f13: Sharpness	A6 f26: Sift FV Probability	A6 f9: Channel Contrast b	A2 f7: Tong_BlurExtent
A1 f55: V DOF Ind	A2 f7: Tong_BlurExtent	A4 f9: Dark Channel	A1 f22: Size feat
A1 f1: mean intensity	A1 f43: P3 Relative Size	A6 f26: Sift FV Probability	A7 f13: Sharpness
A3 f2: Lightning	A8 f147: Eye Sensitivity 7	A4 f18: F2 Shadow area	A3 f10: Color Harmony 6
A2 f1: BBox_Edges	A9 f27: Itten S05	A2 f5: Blur	A1 f1: mean intensity
A2 f9: EdgeDist_Low	A2 f5: Blur	A6 f9: Channel Contrast b	A2 f5: Blur
A2 f5: Blur	A9 f34: Symmetry	A2 f9: EdgeDist_Low	A1 f22: Size feat
A8 f10: Dark Channel 5	A1 f7: ROT mean V	A2 f7: Tong_BlurExtent	A8 f1: Sal Region Area
A4 f9: Dark Channel	A1 f4: mean hue	A8 f92: White Balance 6	A6 f21: Noisiness
A1 f1: mean intensity	A8 f144: Eye Sensitivity 4	A4 f18: F2 Shadow area	A2 f7: Tong_BlurExtent
A1 f54: S DOF Ind	A2 f9: EdgeDist_Low	A2 f1: BBox_Edges	A4 f16: Hue Complexity
A1 f17: V WVT feat L2	A1 f43: P3 Relative Size	A9 f16: Itten H06	A4 f13: Spatial Complexity Fore
A6 f23: Dominant Color a	A1 f55: V DOF Ind	A1 f1: mean intensity	A4 f9: Dark Channel
A1 f7: ROT mean V	A1 f1: mean intensity	A1 f22: Size feat	A9 f16: Itten H06

Table 4.5: List of the top performing features for each of the four image categories of the Hidden-Beauty dataset. Each row shows the algorithm number, based on the order presented in Section 3. and its description. Features on the bottom are the top-performing features without the quality meta-features (NoDLFV).

number of features is ≤ 20, we can see that *people* and *urban* categories observe much smaller drop-off as compared to the *animals* and *nature*. Table 4.5 shows the best performing features for the different categories with (top) or without (bottom) the quality meta-features. As can be seen from the top of Table 4.5, in all categories the DL Probability (probability of being high quality image based on the CNN features) is the most informative feature for predicting continuous aesthetic score.

The bottom of Table 4.5 provides us with information about the type of features that are important to predicting the scores for images in each category without higher-level meta features. It is observed that features describing sharpness or technical quality of the image are important across all of the categories (e.g. features measure blur in [47], Wavelet-based features aiming to capture the Depth of Field (DOF) in [20]). Features capturing different properties of color are observed to be among the most informative for all of the categories (e.g. Mean Hue and Hue Complexity). It is interesting to note that some of the most important features in each category are very intuitive. For example, in the *animals* category, the single most important feature is the first feature described

in [47], which measures the normalized area of a bounding box enclosing 90% of the edge energy in the image. Such a feature is important, since it highlights images with well-defined subjects (animals in the foreground of a blurred background). Often, many of the most appealing images of landscapes and flowers have vivid color. This notion is captured by the best-performing feature for the *nature* category: a bin in the Itten histogram [74] which measures the number of pixels in the image with high saturation. Similarly, many of such images are very symmetric, as is captured by the Symmetry feature, which measures the absolute difference between the Histogram of Oriented Gradients (HOG) descriptors of the left and right halves of the image [94]. Lastly, for the *people* category, some of the most important features measure the shadow area on faces (isolated by a pre-trained face detector) and white balance in the image. This corroborates the notion that lighting of the faces affects our perception of the image [100].

4.4 Combining the CNN and HC features

In this section, we investigate the possibility of improving the performance of deep learning models by fusing hand-crafted features with CNN activations from the penultimate layers of the networks and use them to predict both the mean aesthetics score on the HB dataset and high / low quality images in the AVA dataset. In combining the HC features, we consider the top-performing hand-crafted features after feature elimination based on the assumption that they will provide the most discriminatory power (we examine both inclusion and exclusion of the DL Probability and Fisher Vector features in the respective combinations).

4.4.1 Choosing baseline CNN features

In this section, we describe the comparison of a sample of popular baseline CNN architectures, of which we choose two to be combined with HC features. As can be seen in the recent review by Deng et al. [23], many of the baseline models and proposed architectures are based on the popular AlexNet [55], which first achieved state-of-the-art results on the ImageNet competition. We perform a baseline comparison of the following four models: VGG16 [98], VGG19 [98], ResNet50 [33], and Inception [99]. We provide this comparison in order to choose a strong model to compare against the HC features in the following section and to avoid biasing our results by comparing them to weak baseline models.

Experimental Setup

In order to evaluate the different baseline CNN models, we use the CUHKPQ dataset, where we predict the High/Low quality of the images. In estimating the performance of the algorithms, we utilize 80-20 training-testing splits for classification. In order to better estimate the score, we take

the mean of 20 trials, where we randomly perform a split and calculate the respective metrics. For classification, we report the overall accuracy, defined as

$$\text{Overall Accuracy} = \frac{TP + TN}{P + N}, \tag{4.5}$$

where TP is the number of true positive examples, TN is the number true negative examples and $P + N$ is the total number of images.

VGG16	VGG19	ResNet50	InceptionV3
0.918	0.920	0.936	0.894

Table 4.6: Classification performance of the CUHKPQ dataset on the baseline CNN features for CNN models pre-trained on the ImageNet dataset.

All of the models evaluated were top performers in the ImageNet competition, and thus they provided good baseline results both in terms of classification accuracy and regression. Table 4.6 shows that even baseline features from the penultimate layers of all of the models do a reasonably good job in predicting the quality of images in the CUHKPQ dataset as compared to summarized results in [23], with ResNet50 model achieving the best performance and InceptionV3 achieving the worst (not to be considered further).

4.4.2 Improving CNN performance with HC features

Section 4.3.3 considered a way of combining HC and CNN features by using CNN activations to construct a model to predict the meta-feature, indicating the probability of the image being high quality based on the training set of images in the CUHKPQ dataset. This meta-feature is then considered as one of the HC features. In this section, we explore two ways of fusing hand-crafted features, $\mathbf{X}_{HC}$, and learned CNN activations, $\mathbf{X}_{CNN}$, from the penultimate layer of the network: early (classification / regression) and late (classification) fusion. In early fusion, the HC features are concatenated with the CNN features into a single features representation,

$$\mathbf{X}_{early} = [\mathbf{X}_{HC}; \mathbf{X}_{CNN}],$$

which is then used to learn a function $f : \mathbf{X}_{early} \rightarrow \mathcal{Y}$. Alternatively, we explore a late decision-level fusion by model stacking, where we learn two levels of models (this necessitates splitting the training set features for the particular dataset into two parts). In the first level, we learn separate models f_{HC} and f_{CNN}, based on HC and CNN feature representations respectively (using the first part of each $\mathbf{X}_{HC}$ and $\mathbf{X}_{CNN}$). Then we use the second model

$$f_{stack} : [f_{HC}; f_{CNN}] \rightarrow \mathcal{Y}$$

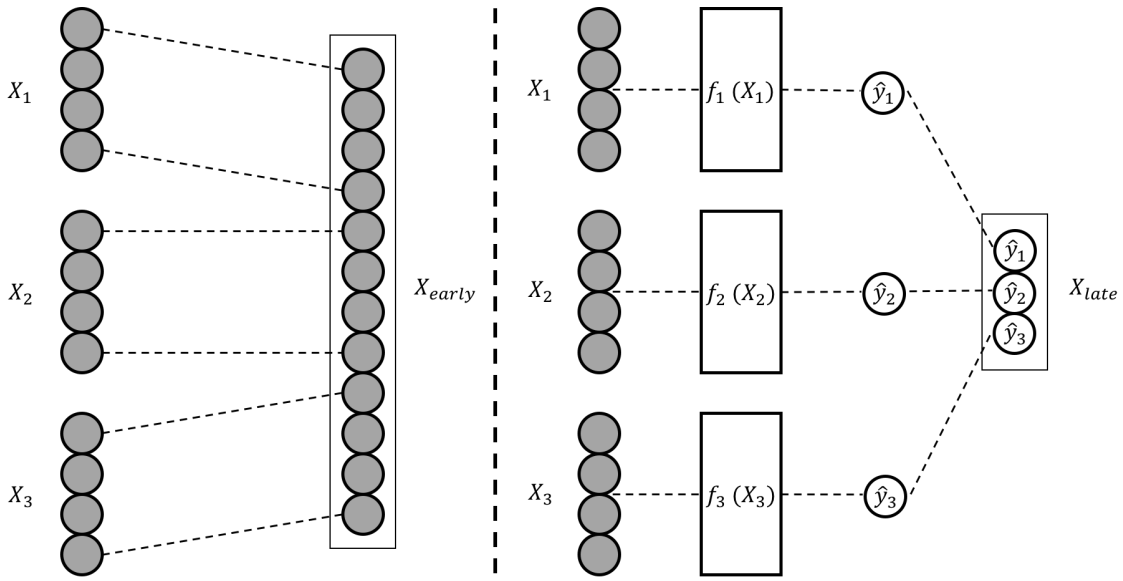


Figure 4.5: Structure of the general pipeline, where we concatenate the HC features with CNN activations.

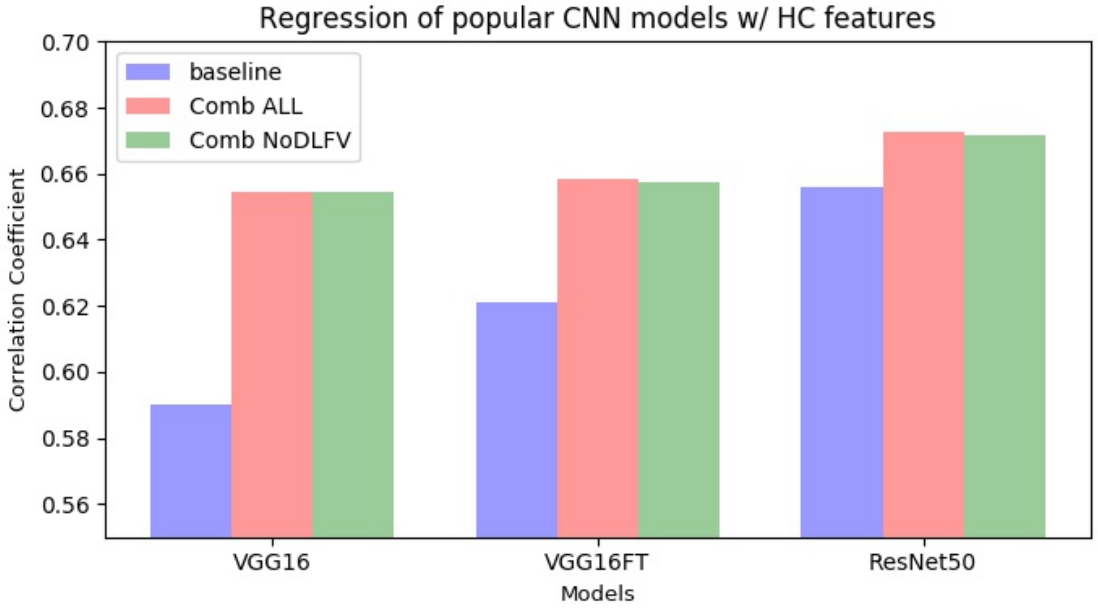


Figure 4.6: Regression performance on the HiddenBeauty score for various CNN models and their combination with HC features.

Previous work	Overall Accuracy
AVA handcrafted features (2012) [80]	68.00
Kao et al. (2016) [46]	74.51
RAPID - improved version (2015)[68]	75.42
DMA net (2015) [69]	75.41
Kao et al. (2016) [45]	76.15
Wang et al. (2016)[106]	76.94
Kong et al. (2016)[53]	77.33
Mai et al. (2016) [75]	77.40
BDN (2016)[108]	78.08
ILGNet (2017) [43]	79.95
VGG16	79.41
VGG16 Early Fusion	80.83
VGG16 Late Fusion	81.65
ResNet50	81.27
ResNet50 Early Fusion	81.79
ResNet50 Late Fusion	81.95

Table 4.7: Classification performance of different models on the AVA dataset.

	VGG	**ResNet**
$\mathcal{H}_1 : \pi_A \neq \pi_B$	$1.176\text{e}{-8}$	0.01482
$\mathcal{H}_1 : \pi_A < \pi_B$	$5.5881\text{e}{-9}$	0.00741

Table 4.8: The following table shows the p-values for the one-sided and two-sided McNemar Test [25] at the significance value of $\alpha = 0.05$

to learn a function to combine the decisions of the first-level models. As we will show, both early and late fusion approaches on average improve the performance in prediction of both the mean aesthetics score and binary classification (see Figure 4.5).

Figure 4.6 shows the summary of results for predicting mean aesthetic score, where the "baseline" R^2 comes only from the CNN features, whereas "Comb ALL/NoDLFV" combine the HC and CNN features[2]. As we can see from our results, simply concatenating HC features and CNN gives a more significant improvement as compared to using CNN features as a meta-feature. Al-

[2]The William's test for the difference between correlated correlations return a value of $t = -5.41$ and $p = 6.2\text{e}{-8} < \alpha = 0.05$ for the regression performance before and after adding HC features to ResNet CNN features, suggesting this improvement is statistically significant.

Model	ResNet	VGG16
1	A2 f6: Tong_Per	A2 f6: Tong_Per
2	A2 f1: BBox_Edges	A6 f19: X Sharpness
3	A8 f126: Color Harmony 4	A2 f1: BBox_Edges
4	A8 f128: Color Harmony 6	A8 f126: Color Harmony 4
5	A9 f34: Symmetry	A8 f128: Color Harmony 6
6	A6 f19: X Sharpness	A6 f20: Y Sharpness
7	A9 f1: Contrast	A8 f140: Color Harmony 18
8	A9 f45: Contrast	A9 f34: Symmetry
9	A8 f5: ROT4	A9 f45: Contrast
10	A8 f4: ROT3	A8 f4: ROT3
11	A6 f17: ColorComp V2	A9 f1: Contrast
12	A8 f130: Color Harmony 8	A8 f132: Color Harmony 10
13	A8 f132: Color Harmony 10	A8 f114: Colorfulness 1
14	A9 f9: Arousal	A8 f5: ROT4
15	A9 f27: Itten S05	A9 f16: Itten H06

Table 4.9: Top 15 performing Hand-Crafted features for the models combining HC and pre-trained CNN features.

though fine-tuning VGG16 to predict binary scores on the CUHKPQ dataset does provide better baseline features, the baseline features are combined with HC features, and we see a very negligible difference between the two models. Additionally, including the meta-features to predict the quality of the images based on CNN and FV provides little improvement. Figure 4.6 shows that ResNet50 achieves better results than VGG16 and achieves a smaller improvement when concatenating it with HC features.

In order to quantify the improvement HC features can provide in a real world scenario, we train a binary classifier based on early and late fusion approaches described earlier to predict the high / low quality of images on the AVA dataset. Similar to strategy used in previous papers [43, 68, 80], the AVA dataset is split into high / low quality images by assigning those images with a score ≥ 5 to the high quality class with roughly 235,000 images being used for training and 20,000 images for testing.

Table 4.7 lists the performance of the various algorithms. Although both of the baseline networks achieve very good performance in predicting the binary aesthetics, an improvement of up to 2.2% can be achieved by fusing the network features with HC features, with decision-level fusion achieving a bigger improvement as compared to feature-level fusion. Table 4.8 shows us the

[2]For the training/testing splits, please see `https://github.com/BestiVictory/ILGnet` [43]

(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.7: Sample images from the AVA dataset. (a) Top correctly classified images of High Quality. (b) Top correctly classified images of Low Quality. (c) Incorrectly classified images of High Quality. (d) Incorrectly classified images of Low Quality. (e) Images of High Quality that were correctly classified by concatenating HC features. (f) Images of Low Quality that were correctly classified by concatenating HC features.

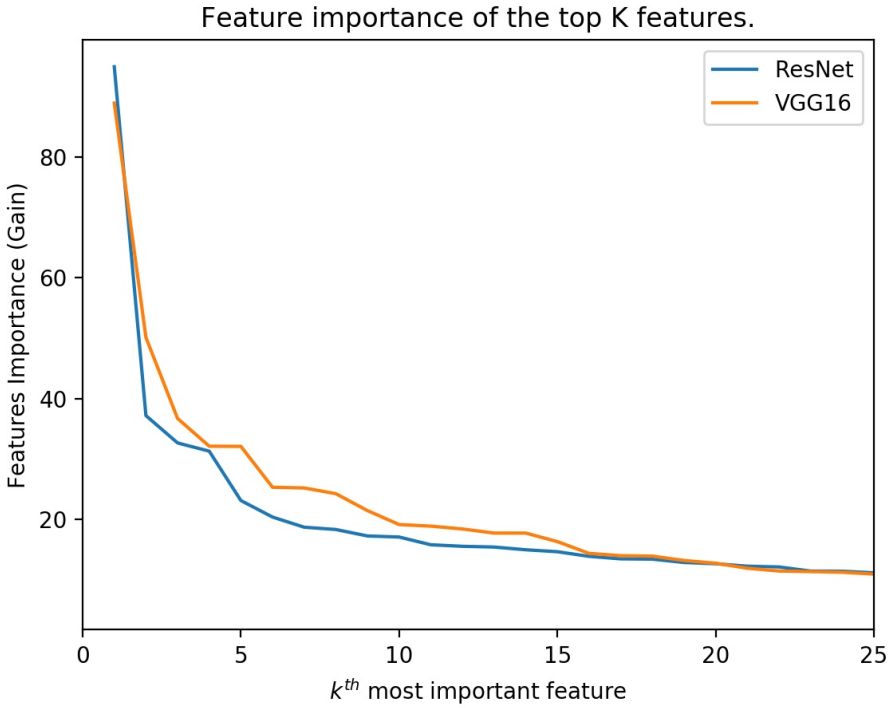


Figure 4.8: Feature Importance (Gain) of the k^{th} feature.

p-values for comparing the classifiers A (baseline) and B (fusion) tested at the significance level of $\alpha = 0.05$. As can be seen, in each case we have that $p - \text{values} < \alpha$, indicating we should reject the null hypothesis $\mathcal{H}_0$ at the 5% significance level. This suggests that the fusion of CNN and HC features does indeed improve the performance of the models.

Figure 4.7 shows sample images that were correctly classified by the models with combined features (Figure 4.7 (a) and (b)), misclassified (Figure 4.7 (c) and (d)), and images classified by the models with the combined features but misclassified by the models based only on the pre-trained features (Figure 4.7 (e) and (f)).

Since each baseline achieves a significant improvement after feature fusion, we can examine the top-performing HC features for the different models and understand, for example, the type of high level knowledge that is approximated by the HC features and missing from CNN, as well as the differences between the various CNN models. Table 4.9 shows the fifteen most important hand-crafted features (note that among the top 100 features as ranked by the gain-based ranking criterion, 30 and 49 features are hand-crafted for the RESNET and VGG16 model, respectively),

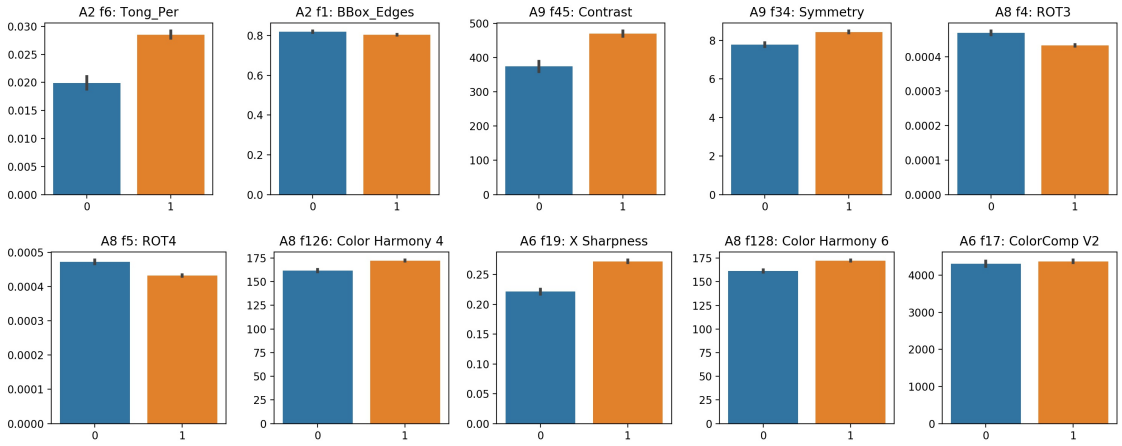


Figure 4.9: Plot of the distribution of the top performing hand-crafted features across the high and low quality classes.

and Figure 4.8 shows the plot of the feature importance as quantified by the gain in descending order for the top features. It can be seen that both of the models use very similar features to improve aesthetics classification, which include features that relate to photographic rules used by professional photographers, e.g. Symmetry features of A9 or features that measure sharpness / blur of photographs. Despite both of the feature sets using features that capture similar information, HC features have a larger impact in improving the performance of the model when concatenated with VGG features as opposed to ResNet features (HC features improve VGG model by 2.2% as opposed to 0.7 % with ResNet).

To gain a better understanding of the images that were correctly classified with HC features, in Figure 4.9 we plot and examine the distribution of the values across the high and low quality classes of the top performing features as judged by the model. The best-performing HC feature for both of the models is the Wavelet-based *Per* feature defined in [104] measuring blur, where an image is said to be un-blurred if *Per* is greater than some threshold. We can see from the first plot that images of high quality indeed have a higher value *Per* and thus are less "blurry". Similarly, *BBox_edges* feature estimates the size of the bounding box enclosing 90 % of edge energy in the image [47]. Intuitively, if the image has a defined subject, most of the edge energy should be concentrated within a smaller box, which is indeed true. The ROT3 feature estimates the normalized distance from the center of mass of a saliency map to the anchor points of an image (see Figure 4.1). As we can see, the images with higher quality tend to have lower distance to one of the anchor points, suggesting a better adherence to the rule of thirds in photography composition.

Future work and limitations

Our work in this paper has focused on improving the predictive performance of deep learning features by their fusion with hand-crafted features. We showed that such fusion can improve the performance over baseline prediction on several datasets and paradigms (classification vs. regression). One of the factors that could be explored in the future work is the influence of the number of training examples on the marginal improvement of fusing hand-crafted features over the baseline deep features. We saw an improvement in predictive performance on all 3 of datasets: Kodak Aesthetics dataset, HiddenBeauty dataset, and AVA dataset with rouhgly 1500, 15 000, and 250 000 examples respectively. From the Universal Approximation Theorem (UAT) [19] we know about the expressive power of neural networks. The UAT states that "a neural network with a single hidden layer can approximate arbitrarily well a continuous function". Thus it could be expected that as the number of examples available to us approaches infinity the marginal improvement in performance by fusing the neural network features with hand-crafted features would approach zero. Though this is a valid concern, one of the main problems in many domains is often the lack of large datasets.

4.5 Conclusion

In this chapter, we studied and compared a selection of algorithms that use hand-crafted features designed to assess image aesthetics. We show that even early algorithms can provide adequate results in their efficacy of predicting image aesthetics as compared to more recent methods based on hand-crafted features. We can achieve an additional improvement in aesthetic prediction accuracy by combining all of the features together and attain a performance close to that of a model trained on learned CNN features. By performing feature elimination, a good performance for classification / regression can be achieved for a specific combination of just 25 and 40 features respectively out of more than 300 features. Furthermore, we can see that even if we remove a large portion of features (in our case these were features from Algorithm 8), we achieve a very similar performance with features from different algorithms that captured very similar information (e.g. Color Harmony from A4 vs A8). By analyzing the combination of all features on the different categories, we find that the most important features of each category are intuitively important for each of the categories. Furthermore, when fusing HC features with pre-trained deep learning features, we can achieve a significant improvement in predicting both a continuous aesthetic metric, and predicting a binary high / low quality score with improvement up to 2.2% in classification accuracy.

Appendix

The McNemar's Test [3] is a hypothesis test for comparing populations proportions, considering the fact that the data come from two dependent matched-pair samples [25] (i.e. the predictions of the classifiers are trained and tested on the same training / testing splits). Assuming a learners A and B and their corresponding decisions functions $f_A(x)$ and $f_B(x)$ were trained on the same training set, let $\{\hat{y}_i^A\}$ $\{\hat{y}_i^A\}$ be the predictions for the test set obtained from the learners A and B respectively. Then the two-sided test for comparing the accuracies of the models is:

$$\mathcal{H}_0 : \pi_A = \pi_B$$
$$\mathcal{H}_1 : \pi_A \neq \pi_B$$

where π_i represents the misclassification rates of the two models (in our case, the model A corresponds to the model trained will solely deep features B is the model that combines the CNN features with HC features). Alternatively, we can test the the the alternative hypothesis $\mathcal{H}_1 : \pi_A < \pi_B$.

[3] See MATLAB function *testcholdout* for an implementation of the algorithm.

Chapter 5

Aesthetic Inference for Smart Mobile Devices

In the previous chapter, we explored hand-designed features and their possibility in aiding baseline deep learning features in predicting image aesthetics. Although, the raw performance is often of interest, in this chapter we focus on quantifying how well we can predict image aesthetics in the case where we have limited computational resources on devices such as cell phones. We show that even with constrained resources, we can achieve adequate results in aesthetic ranking, which can also be used for aesthetic cropping. Aesthetic ranking networks are shown to perform near state-of-the-art as compared to aesthetics-based image croppers.

5.1 Introduction

Mobile phones and their cameras have come a long way in the last fifteen years. Nokia 7650, one of the first camera phones released in the June 2002, featured a 4MB of memory and a 0.3 megapixel rear camera which captured images with a resolution of 640×480 pixel [112]. This is a far cry from the cameras that we see today, such as the rear camera of the recently released iPhone X, which features a 12 megapixel dual-lens camera capable of shooting 4K video [111]. Additionally, the phones have seen exponential increases in the available storage memory with phones often starting at 64 GB, e.g. the OnePlus 5 or the iPhone 8, allowing for users rarely having to transfer and manage the media on their phones. This allowed the smart-phones to become one of the main media through which people capture and share their daily lives and special moments such as outings with friends or birthdays, resulting in thousands of photos being stored on each user's phones, for examples see Figure 5.1. This gives rise to several problems, such as memory management or photo curation.

(a) Born to be Wild[1]

(b) Pets[2]

Figure 5.1: Examples of photographic images takes by cell-phones.

In addition to the significant improvement in the camera quality, we have seen the incredible increase in the computational power present in mobile handsets, e.g. the Apple A11 processor which achieved better GeekBench benchmark scores as compared to a recent laptop Intel processor [59]. This impressive feat, along with special purpose hardware (e.g. the Visual Processing Unit of the Google Pixel 2), can allow us to run more sophisticated algorithms whether it is for image intelligence or for image processing itself.

Determining aesthetic appeal is a challenging problem since it relies on various image properties (e.g. technical quality, composition, color contrast) as well as context (i.e. presence of other images). Despite its challenging nature, aesthetic inference has made significant progress due to the emergence of large labeled datasets (e.g. AVA dataset [80]) enabling learning based approaches. Such datasets are described in Section 2.2.1. In this study we aim to understand how well CNN models perform aesthetic inference under constrained resources. The ability to do aesthetic inference on our phones could enable algorithms that better organize our photo-albums, automatically remove bad images or predict image enhancements (e.g. cropping). Although both global and local details are very important in judging image aesthetics, as Joshi et al. [44] point out, we do not necessarily consider individual elements of an image rather we view it as a whole. More specifically, we utilize the MobileNet [36] architecture, and investigate how varying the size of the input images and α (the parameter controlling the layer depth), affect the performance of these architectures in terms of their ability to do binary classification of images into High / Low quality categories, ranking images, and using them as part of an image cropping algorithm. To the

[2]by Margarita Iskandarova. Source: Mobile Photography Awards
[2]by David Pierce. Source: Wired

best of our knowledge, our study is the first to investigate the performance of aesthetic inference for models aimed at mobile devices. Our main contribution is three-fold:

- We train sixteen different variations of the MobileNet architecture for aesthetic inference
- We compare the effect of image size and layer depth of the performance of the models.
- We report the results for two standard benchmark datasets: AVA dataset [80] and Flickr Cropping Database [15]

The rest of the chapter is organized as follows. In section 2. we briefly review the related work on efficient neural network models. Section 3. and section 4. discuss the MobileNet architecture and and its modification for aesthetic inference and evaluation. In section 5. we discuss the results of predicting aesthetics using various MobileNet architectures and demonstrate their use for image cropping.

5.2 Related Work

Despite the dominance of deep learning methods in state-of-the-art computer vision, they are still mostly used with specialized hardware (e.g. Graphical Processing Units). A desire to run these models on smaller devices due to their strong performance on various computer vision tasks resulted in research trying to build smaller and more efficient models, mainly focusing on either compressing larger models, training smaller models and creating architectures with more efficient operations.

MobileNet is an efficient architecture based on Separable Convolution operation, which was previously used in the early layers of Inception-V3 [99], and later in the Xception architecture as its major building block, achieving then state-of-the-art results. Flattened Networks further factor out the 2D filters into two separate horizontal and vertical filters [42]. Some of the other approaches used to create more efficient networks are by parameter pruning [62, 78] , factorization [39], quantization [105] or knowledge distillation [35]. It is interesting to note that these approaches are complimentary, and thus could be further used to optimize the latency of the networks running on mobile devices.

5.3 Methodology

Early aesthetic inference algorithms mostly focused on predicting the aesthetics of an image by trying to assign it to a High or Low quality category. However, in the case of a photo collection one is more interested in the relative image quality or beauty. Therefore we treat the problem as a joint ranking and regression problem. We use the MobileNet architecture recently proposed

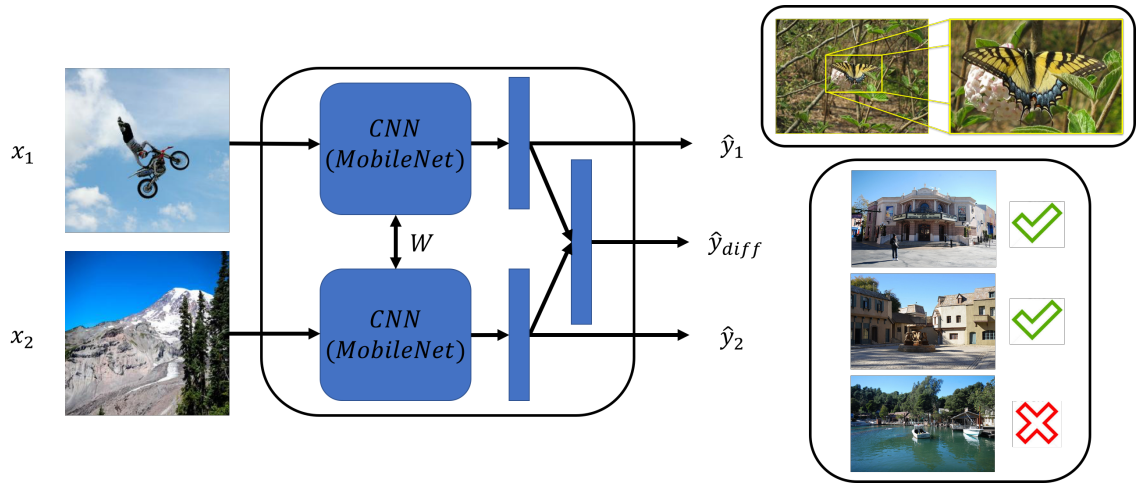


Figure 5.2: Figure showing the high-level architecture of our model with the multiple outputs (left), and possible uses for such model (right).

by Howard et al. [36]. Figure 5.2 provides the architecture that was optimized for aesthetics inference and some envisioned uses of the trained models (top shows image cropping and bottom photo album curation). We first provide a brief overview of the MobileNet architecture and then go on to describe the modifications and training methodology in the subsequent sections.

5.3.1 MobileNet architecture

The base network architecture presented in [36] takes as an input a batch of images (each image is normalized to have input values between -1 and 1) and applies standard convolutional layer to the input. The output of the first layer is further processed with 13 depthwise separable convolutions. A global average pooling layer is applied to the last convolutional feature map to reduce image representation, followed by a fully connected layer, whose shape is determined by a given task (e.g. 1000 neuron softmax layer for ImageNet classification). The depthwise separable convolution is composed into two separate operations: a depthwise convolution applying a separate $n \times n$ filter to each of the m input channels, followed by a 1×1 convolution which serves to create new features. This decomposition results in reduction of computation by up to 9 times with a minor loss in accuracy [36]. In order to control the computational cost and number of trainable parameters in the architecture, Howard et al. introduce two parameters [36]. A width multiplier, α, which controls the number of input and output channels of individual layers, has direct influence on both the computation and the number of trainable parameters in the networks. The resolution multiplier, ρ, modulates the size of the input and only affects the computational cost of each model at the risk

of reducing the amount of information available to the model.

5.3.2 Multi-task Training

Much of the previous work on predicting image aesthetics focused on distinguishing between "good" and "bad" images, which however could make it more difficult and arbitrary to assign images near boundaries of the classes in datasets (i.e. arbitrarily defined boundaries created by thresholding a continuous score). Inspired by Kong et al. [53], we optimize the network to both predict the absolute aesthetics score, as well as to predict the relative ranking of two images by optimizing both a regression and ranking loss, but we do not use additional layers to predict either aesthetic attributes or content of the images.

Regression Loss

We first aim to predict the continuous aesthetics score for each image by minimizing the l_2 loss between the predicted score, $\hat{y}$, and the ground truth score, y,

$$l_{reg} = \frac{1}{N}\sum_{i=1}^{N}\|y_i - \hat{y}_i\|^2. \tag{5.1}$$

Ranking Loss

When looking at images, humans often judge their appearance on a relative scale by comparing them against other images, for example when trying to pick the best images out of an album. To improve the fine-grained aesthetics inference, we include the hinge ranking loss given by:

$$l_{rank} = \frac{1}{N}\sum \max(0, \xi - \delta(y_i^1 \geq y_i^2)(\hat{y}_i^1 - \hat{y}_i^2)) \tag{5.2}$$

where $\delta(y_i^1 \geq y_i^2)$ is 1 if $y_i^1 \geq y_i^2$ otherwise it takes the value of -1, and ξ is specifies the margin parameter.

Difference Loss

Inspired by [107], in addition to the regression and ranking losses, we add an additional pathway to predict the difference in aesthetic scores between two images. The feature activations from the two images I_1 and I_2 from the pen-ultimate layer of each column are concatenated together and fed into a fully-connected layer to predict the difference $\hat{y}_{diff}$ between the aesthetics scores $d = y_i^1 - y_i^2$. We use the euclidean l_2 loss to optimize it:

$$l_{diff} = \frac{1}{N}\sum_{i=1}^{N}\|y_{diff} - \hat{y}_{diff}\|^2. \tag{5.3}$$

Therefore the total loss used to optimize the network is

$$l_{total} = a_1 \cdot l_{reg} + a_2 \cdot l_{rank} + a_3 \cdot l_{diff}, \tag{5.4}$$

where a_i are un-normalized constants to control the trade-off between different training objectives.

In order to facilitate training pairs of images, we employ the Siamese network architecture [11] with shared weights, where the base column is the MobileNet architecture. For each of the columns we remove the top layer originally used for classification and use features from the penultimate layer to predict the absolute aesthetics score for the images. Furthermore, the features from each column are concatenated as an input to another fully-connected layer used to predict the difference score between two images. At test time, we can either decide to use both of the columns to directly predict the score difference, or for more efficient processing, we can simply eliminate the extra branch predicting a difference in aesthetic scores and use only one column of the Siamese network to predict the aesthetics score.

5.4 Experimental Setup

5.4.1 Evaluation Datasets

The *Aesthetic Visual Analysis* (AVA) dataset is one of the largest datasets available for working with aesthetic preferences with more than 250,000 images[80]. The images were sourced from *www.dpchallenge.com*, a website housing a community of amateur and professional photographers. Each image in the dataset received between 78 and 549 votes, with an average of 210 votes per images. Each image is given a score on scale $1 - 10$ and then the average rating is considered to be the ground truth aesthetic score for the image. Along with aesthetic ratings, images come with 66 semantic and 14 photographic style annotations (e.g. High Dynamic Range, Soft Focus, etc.).

The *Flickr Cropping Database* (FCDB) dataset is a recently introduced dataset for evaluation of image cropping algorithms [15]. In order to improve the generalization of the dataset, they collect images in a wider quality range (previous datasets were slightly biased towards high-quality photograph). The dataset contains a total of 1,743 images and provides a ground truth image / crop pair as well as sets of ten crop pairs for each image and their relative ranking.

5.4.2 Training details

The MobileNet architecture was implemented using the TensorFlow [1] Neural Network library. The base-column models were initialized with weights pre-trained on ImageNet dataset, and the fully-connected layers for predicting aesthetic scores and difference were initialized with random uniform distribution with mean of 0 and standard deviation of 0.1. Each model was trained for

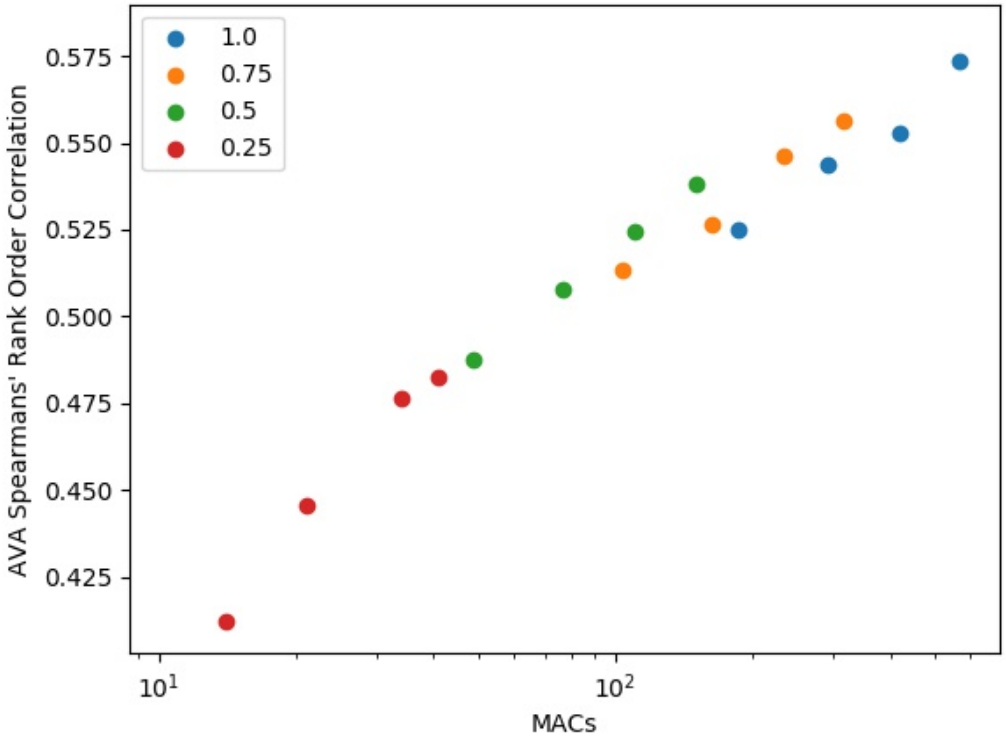


Figure 5.3: Figure showing the tradeoff between the rank-order correlation for the AVA dataset and computation efficiency of individual models (measured in millions of multiply-accumulates, MACs). The points with the same color are models that share same width multiplier. The increase in performance in models of the some color is result of increasing image size.

30,000 steps, optimized using the Momentum Optimizer with a learning rate of 0.0001 and momentum of 0.9 with a batch size of 128. Similar to [36], we choose $\alpha \in \{0.25, 0.5, 0.75, 1.0\}$ and ρ such that the resulting image sizes are $\{128, 164, 192, 224\}$. For each image we take its center crop, resize the image to 256×256 pixels, randomly apply image rotation, scaling and take a random 224×224 sub-crop from the center-crop. For the columns, we do not use any specific sampling technique - we randomly choose 128 images and split them into two batches of 64 images and feed them into the different columns.

5.4.3 Performance evaluation

In order to understand the performance of the different models on the AVA dataset, we report the Spearman's Rank-Order Correlation coefficient, which can be calculated as

$$r_S = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}, \tag{5.5}$$

where $d_i = u_i - v_i$, and u_i and v_i represent the rank of the i^{th} ground-truth and predicted scores respectively [123]. Furthermore, in order to compare the results of the AVA dataset to previously published methods results, the predicted aesthetics scores $\hat{y}$ are assigned to high quality class if $\hat{y} \geq 5$. Thus for the AVA dataset we also report the binary accuracy defined as

$$\text{Overall Accuracy} = \frac{TP + TN}{P + N}, \tag{5.6}$$

where TP is the number of true positive examples, TN is the number true negative examples and $P + N$ is the total number of images.

5.5 Results

In this section we study the effect varying the input size and width multipliers on aesthetics inference using the AVA datasets and explore using the models for image cropping. Similar to previous papers, we use the AVA train / test split with roughly 235,000 and 20,000 images respectively.[3] We further subdivide the training set into train / validation splits for the purpose of parameter tuning.

5.5.1 Aesthetic Inference

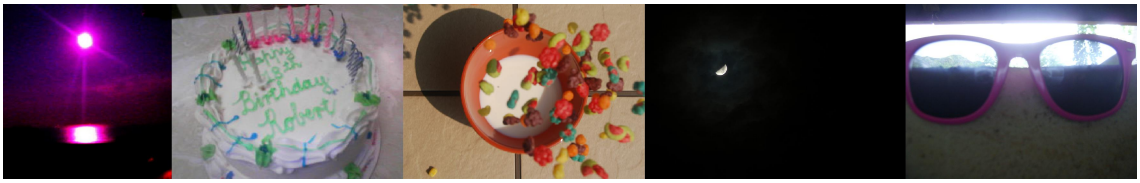
We first study the impact different combinations of loss functions have on the full model. The various model configurations will be denoted as *MobileNet-resolution multiplier-depth multiplier*, e.g. MobileNet-192-0.75 would be a model configuration where input image size is 192 at each side and depth multiplier (controling model complexity) is set at 0.75. Table 5.2 compares the ranking correlation of the the full MobileNet model trained with different loss combinations, where each used the same training procedure described earlier. From the results in Table 5.2 for the different loss combinations, we can see that the difference loss ("DifF") was able to significantly improve the ranking of the images as compared to the ranking loss ("Rank"). In combination, they improve the ranking correlation even further giving us ranking correlation of 0.5735.

[3]The particular training/testing splits are the same as ones used in [43] and can be found at: `https://github.com/BestiVictory/ILGnet`

(a) Best images as predicted by the MobileNet-128-0.25 model.

(b) Worst images as predicted by the MobileNet-128-0.25 model

(c) Best images as predicted by the MobileNet-160-0.5 model.



(d) Worst images as predicted by the MobileNet-160-0.5 model

(e) Best images as predicted by the MobileNet-224-1.0 model.

(f) Worst images as predicted by the MobileNet-224-1.0 model

Figure 5.4: Examples of the best and worst images as predicted by the following models: MobileNet-128-0.25 (lowest performing), MobileNet-160-0.5, and MobileNet-224-1.0 (best performing).

Model Name	Overall accuracy
AVA handcrafted features (2012) [80]	68.00
Kao et al. (2016) [46]	74.51
RAPID - improved version (2015)[68]	75.42
DMA net (2015) [69]	75.41
Kao et al. (2016) [45]	76.15
Wang et al. (2016)[106]	76.94
Kong et al. (2016)[53]	77.33
Mai et al. (2016) [75]	77.40
BDN (2016)[108]	78.08
ILGNet (2017) [43]	82.66
A - Lamp (2017) [73]	82.50
MobileNet-224-1.0	82.82

Table 5.1: Comparison of the classification results on AVA dataset as compared to previous method as quantified by the binary accuracy.

Model Name	Spearman's ρ
Reg	0.5416
Reg + Rank	0.5495
Reg + Diff	0.5599
Reg + Rank + Diff	0.5735

Table 5.2: Comparison of performance in ranking the AVA dataset of the MobileNet-224-1.0 trained with different losses.

Model Name	Spearman's ρ
MobileNet-224-0.25	0.4823
MobileNet-224-0.5	0.5382
MobileNet-224-0.75	0.5562
MobileNet-224-1.0	0.5735

Table 5.3: Comparison of the effect the width multiplier has on aesthetic ranking.

Model Name	Spearman's ρ
MobileNet-128-1.0	0.5251
MobileNet-160-1.0	0.5437
MobileNet-192-1.0	0.5528
MobileNet-224-1.0	0.5735

Table 5.4: Comparison of the effect the resolution multiplier has on aesthetic ranking.

Model	Overlap	Disp.
SVM + $DeCAF_7$ [15]	0.5154	0.1325
AVA 1-1+$DeCAF_7$ [15]	0.5223	0.1294
CFDB+SIFT-FV [15]	0.5917	0.1084
CFDB+$DeCAF_7$ [15]	**0.6019**	**0.1060**
MobileNet-128-0.25	0.5150	0.1315
MobileNet-128-0.50	0.5562	0.1208
MobileNet-128-0.75	0.5393	0.1263
MobileNet-128-1.0	0.5726	0.1156
MobileNet-160-0.25	0.5083	0.1358
MobileNet-160-0.50	0.5275	0.1285
MobileNet-160-0.75	0.5663	0.1170
MobileNet-160-1.0	0.5642	0.1180
MobileNet-192-0.25	0.5519	0.1213
MobileNet-192-0.50	0.5413	0.1247
MobileNet-192-0.75	0.5631	0.1190
MobileNet-192-1.0	0.5840	0.1120
MobileNet-224-0.25	0.5648	0.1182
MobileNet-224-0.50	0.5967	0.1076
MobileNet-224-0.75	0.5677	0.1174
MobileNet-224-1.0	0.5628	0.1190

Table 5.5: Comparison of the MobileNet architecture in their ability to pick the best crop as compared the models in [15].

Table 5.1 compares binary classification accuracy (High / Low quality) of the best MobileNet model to the previously published models . As we can see in Table 5.1, although the full MobileNet model ($\rho = 1.0$ and $\alpha = 1.0$) is able to achieve results competitive to the state of the art result of [43, 73], it also significantly reduces computation, since for example [73] use VGG16 network to process multiple image patches (VGG16 is 32 times larger in the number of parameters and performs 27 times more compute operations [36]).

5.5.2 Image Cropping

Often after taking pictures we desire to crop the image in order to improve its appearance by removing undesirable or redundant areas of the image. In this section we examine the possibility of using the MobileNet models trained for aesthetic inference for the purpose of image cropping. Fol-

lowing the evaluation protocol of [15], sliding windows with multiples of $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ the original size are used to evaluate the crops on a 5×5 regular grid to pick the best image crop. Each crop is pre-processed and fed into the specific MobileNet model to obtain its aesthetics score, and then the crop with the highest aesthetics score is chosen as the best image crop. Similar to [15], of each model we report the average overlapped ratio, defined as

$$\frac{1}{N}\sum_{i=1}^{N} area(W_i^g \cap W_i^c)/area(W_i^g \cup W_i^c), \tag{5.7}$$

where W_i^g and W_i^c are the ground truth and predicted crop windows, and the boundary displacement error, defined as

$$\sum_{j=l,r,b,u} \|B_j^g - B_j^c\|/4, \tag{5.8}$$

where B_j^g and B_j^c are the ground truth and predicted crop edges.

Table 5.5 compares the performance of the various MobileNet architectures in their ability to pick out the best image crops. Although none of the models beat the best results from [15], most of the models do better than their aesthetics and saliency based methods. This could be explained by the way our models were trained - the ranking and difference loss encourage the fine-grained ranking. This is not true of the aesthetics based model trained on AVA dataset in [15], which only trained models which used the best and worst rated images from the AVA dataset. Figure 5.5 shows the best crops from the different models as compared to the ground truth. Although the aesthetics-based croppers using MobileNet models perform on par with state-of-the-art cropping models based on image aesthetics, the do not perform as well as models whose aim is to model image composition.

Figure 5.3 plots the performance of the different models as a function of number of Multiply-Accumulates (MACs). Similarly to [36], we observe the log-linear relationship of Spearman correlation and MACs. In Figure 5.4 we show the best and worst images from the test set as predicted by various models. Table 5.4 shows the comparison of the Spearman's rank order correlation for the AVA dataset as a function of input size. and Table 5.3 shows the Comparison of the Spearman's rank order correlation for the AVA dataset as a function of width multiplier. As expected, decreasing either of the multipliers results in a decrease in the ranking correlation.

5.6 Conclusion

In this chapter we studied the problem of aesthetic inference with models optimized for efficient computation. We study the effects of the width multiplier, resolution multiplier, different loss combinations and their effect on the Spearman's rank-order correlation. We demonstrate that the joint combination of the regression, ranking and difference losses achieves the best performance in

(q) Ground Truth Annotation (r) MobileNet-224-1.0 (s) MobileNet-224-0.5 (t) MobileNet-128-0.25

Figure 5.5: Examples of best image crops predicted by different models as compared to ground truth.

aesthetic ranking. Using the combination of losses, we see that the full MobileNet architecture can achieve near state-of-the-art results in binary classification accuracy while using only a fraction of the computation, and serve as a better proxy for picking the best image crops as compared to other aesthetics-based croppers.

Chapter 6

Learning representations for composition ranking

Though in previous chapters we have shown that aesthetic ranking functions can achieve very good results as compared to other aesthetics-based croppers, baseline composition ranking functions are known to achieve better results showing that aesthetic ranking functions do not have perfect knowledge of composition. Therefore, in this chapter we set out to identify the key choices that allow for training of state-of-the-art composition ranking functions.

6.1 Introduction

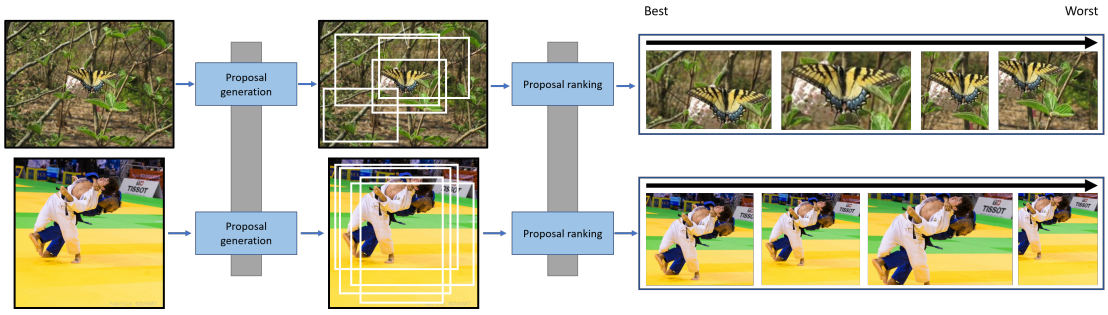


Figure 6.1: A Figure that illustrates image cropping as a two stage process

Automatic image cropping algorithms find use in various applications such as thumbnail generation [102], view finding [110], and image enhancement, though they all share a common goal:

(a) Aesthetics image pairs

(b) Cropping image pairs

Figure 6.2: Figure that illustrates the difference in image-pairs that are used to train (a) aesthetic, and (b) composition ranking functions

to remove a part of the image background to better suit the desired outcome. Various approaches were developed for this purpose including ones based on saliency [101], aesthetics [61], or view ranking based on composition or aesthetics [16, 58]. Thumbnail generation focuses on generating a smaller version of the picture in order to conserve space, and thus an approach as taken by [102] seems natural, which utilizes a saliency algorithm [101] to highlight the most "interesting" part of the image. However, such an approach may not be the method of choice in case we are not constrained by size or aspect ratio of the image. When we are seeking the most pleasing sub-view of the image, various approaches have been developed. In particular, we are interested in approaches illustrated in Figure 6.1 which use a function, approximated by a deep network, to rank a set of candidate crops of an image.

Though the notion of aesthetics itself is often hard to pinpoint, people have turned to data-driven methods to try and capture image aesthetics by learning to predict these scores based on some image features (e.g. expert features [57] or pair-wise image ranking [58]). Many works also focus on modeling a related yet different notion of image quality: its composition. Composition is often used to crop images and improve their aesthetics and quality (note that composition is often an important aspect of aesthetically pleasing images). Though recent approaches have shown that aesthetics ranking functions trained through pair-wise learning could achieve near state-of-the-art in aesthetic image cropping [58], such a function is still less effective than methods that model composition. This can be explained by differences in data used for learning aesthetic and composition ranking functions as we can see in Figure 6.2.

Modeling of image composition has been of interest to many fields, especially for the purpose of image enhancement and aesthetics inference. Recently, Zeng et al. [119] released the GAIC dataset, which contains a set of roughly 90 rank ordered sub-views for 1250 images, allowing for data-driven composition modeling and computing more appropriate ranking focused metrics. In our work, we seek to train a ranking network with the goal of learning a function that implicitly

learns an appropriate feature representation for composition and could provide a score for describing the composition quality of the image. Such a ranking function is of interest due to its many uses for (a) image cropping as part of a proposal-ranking pipeline, (b) as a teacher function to provide soft labels [110], or (c) a feature extractor to describe image composition.

Function approximation and representation learning using deep networks have dominated recent approaches to many domains in computer vision. Though in the case of image ranking optimization, the functions can be notoriously hard to optimize. For example, if we consider the triplet loss used for optimizing the representations for image retrieval, much of the improvement comes from training the network with an auxiliary loss (fine-grained classification) [28] to improve initial representations or propose various methods to improve the sampling methods for triplets to improve what examples the network is shown [116]. Therefore, in this work we aim to identify a set of key practices to adopt for learning representations that capture image composition. We aim to do this, as previous work has used various models, however some have neglected to consider the various choices that could have impact on the performance of the model, e.g. image size or the type of pooling mechanism the network uses.

The remainder of the Chapter is organized as follows. In Section 2. we summarize previous work on modeling image composition and its uses, and various approaches for modeling data-driven composition. Section 3. explores the various choices critical for learning ranking functions for image composition. Section 4. discusses the results of the ablation studies and comparison to state-of-the-art models. Concluding remarks and suggested future work is discussed in Section 5.

6.2 Related work

Composition modeling for aesthetics.

Early work on modeling image composition focused on coming with up hand-crafted features that capture rules of composition from photography, such as the rule of thirds [20], or the symmetry of the image [94]. These metrics were then used as descriptors for image cropping [117] or aesthetics prediction [72]. For example, Luo et al. [72] define a "Composition Geometry Feature" that approximates the rules of thirds by computing the minimum distance from the center of mass for a salience map to the "power points" of the image [85]. Dhar et al. [24] predict whether the image satisfies the rule of thirds based on a set of hand-crafted features, which are then used as meta-features to predict aesthetic quality. Zhang et al. [120] segment the image into regions, and use a graph-based approach to model the spatial structure of the image and understand its composition to predict the aesthetic quality of the image. More recently, several models used composition to motivate their choices in designing their deep learning approaches to image aesthetic assessment. Liu et al. [64] partition the image into a region composition graph to model its composition, and use a graph convolution network to evaluate the aesthetic quality of the image.

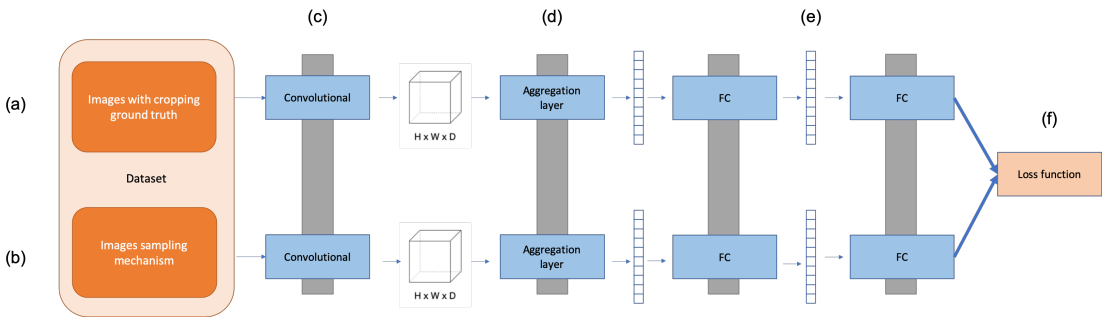


Figure 6.3: Schematic of architectures considerations for the ranking model.

Data-driven composition modeling.

Recently, several works became interested in modeling image composition through deep learning. Chen et al. [16] proposed to learn a ranking function from professional photographs. They draw attention to the fact that professional photographs often have high composition quality. To make use of this, they propose to take various sub-crops and use them in conjunction with the original image to train a Siamese network [11], in which they assume that the sub-crop is of worse composition. Wei et al. [110] introduce two datasets, from which they harvest image pairs for training the View Evaluation Network (VEN). VEN is used as a teacher network to train a region proposal network for generating pleasing sub-views of the image. Debang et al. [61] propose a sequential approach for training a reinforcement learning agent, which is penalized based on the aesthetic quality of the image. Zeng et al. [119] recently released a dataset, whose aim is to directly model image composition, where each image comes with a rank ordered set of sub-views and thus can be used to compute ranking focused metrics. Additionally, they present a cropping-focused model, in which the image and set of bounding boxes (originating from a regular grid) are fed into an object detection-like framework to predict a composition score for the various crops.

In our work, we aim to challenge the claim of Zeng et al. [119] that pair-wise ranking methods are inferior in learning composition ranking functions.

6.3 Method

In this section we outline our general approach to composition ranking, and its application to image cropping. Though one can approach image cropping in several ways, we choose to view it as a two-stage problem of (a) proposal generation and (b) proposal ranking. Our work focuses on improving the image ranking function for image composition. Previous deep learning approaches

that used CNN-based ranking functions, used various approaches to mine pair-wise image views. As Zeng et al. [119] demonstrate recently, they fall behind in their composition ranking ability as measured by average Spearman's Rank Order Correlation ($\overline{SRCC}$) and Accuracy ($\overline{ACC}$). Zeng et al. utilize an approach rooted in object detection to obtain scores in which they considered both the regions that are of interest and those that are discarded. Previous CNN ranking approaches only used architectures based on AlexNet [55] and VGG16 [98], despite the existence of stronger baselines.

The detection based approach [119] utilize the VGG16 and ResNet50 [33] architectures as feature extractors, however report worse performance on the ResNet50 architecture. This is contrary to intuition, as several works have shown that better architectures (smaller Top-5 error rate on the ImageNet dataset [22]) tend to perform better in various tasks. Konblith et al. [54] show through a comparison of 16 networks that better models achieve better transfer learning performance. Similarly, Kucer et al. [57] show that expert features provide smaller marginal improvement in combination with ResNet features as opposed to VGG16. Gordo et al. [28] show better models perform better for image retrieval as well.

Representation learning and ranking optimization are notoriously difficult to optimize [28]. Our exploration takes inspiration from Gordo et al. [28], which shows the nuance and difficulty of optimizing CNNs for image retrieval. In the following sections we outline the architectural choices and training strategy that are critical to training composition ranking models no matter what backbone one chooses to use based on a purpose they are trying to pursue (performance vs efficiency). In Figure 6.3 we outline the various parts of the architecture that are under consideration.

6.3.1 Weight initialization.

Previous work by Gordo et al. [28] found that a classification pre-training step was critical in improving the representation learning and convergence of the triplet loss for image retrieval. In this step, an auxiliary task of fine-grained classification was used to fine-tune the network weights, before they were used to initialize the Siamese network for learning a representation. Our first experiments explored ways of pre-training the network weights with auxiliary losses on similar tasks. In particular, we pre-trained the networks for aesthetic prediction using binary and multi-class aesthetics classes, and binary prediction of composition quality. In all cases, we found that this step was detrimental to the ranking performance of the model. Therefore, all of our ablation studies are initialized with weights pre-trained on the ImageNet dataset [22].

6.3.2 Architecture design

In Figure 6.3, we see that the major parts of the architecture resemble standard deep learning models for object classification or ranking. The pipeline consist of the following high-level parts:

(a) dataset, (b) sampling mechanism, (c) the backbone and knowledge of presence of batch-normalization within the network, (d) pooling mechanism and the dimension of convolution features, (e) top-level classifier / ranking network, and (f) the loss function. We consider these parts of the architecture as these constitute the major architectural differences between AlexNet / VGG16 and MobileNet V2 / ResNet50. First, the backbone consists of the convolution layers from any architecture (VGG16, ResNet, Inception), whose weights have been pre-trained on ImageNet. The backbone takes as an input a transformed image and outputs a set of convolutional features maps. These feature maps are fed into an optional 1×1 convolution layer, which will be used to reduced the size of the feature vector at each spatial location. The resulting feature maps will then be fed into a pooling / flattening layer. In this case we use either max-pooling, mean-pooling or simply flattening the convolutional feature maps into a $1 - D$ vector. These outputs will the be fed into a fully-connected layer of size f, which will be used to predict a single composition score.

6.3.3 Learning composition ranking.

Though many successful approaches for image retrieval use the triplet loss, it is harder to formulate the composition ranking in the same paradigm as there is no good definition of what would constitute the query, positive and negative examples. Therefore, similar to previous aesthetic and composition ranking approaches [58, 110], we will optimize a pair-wise ranking loss of the form

$$L(I_i^1, I_i^2) = \max(0, \xi - \delta(y_i^1 \geq y_i^2)(\hat{y}^1 - \hat{y}^2)), \tag{6.1}$$

which will be used to compute the loss for a batch of pairs

$$l_{rank} = \frac{1}{N} \sum_{i=1}^{N} L(I_i^1, I_i^2) \tag{6.2}$$

where $\delta(y_i^1 \geq y_i^2)$ is 1 if $y_i^1 \geq y_i^2$ otherwise it takes the value of -1, ξ specifies the margin parameter, and y_i^1 and y_i^2 are the ground truth composition scores for image 1 and image 2. To obtain the scores, pairs of images will be fed into a two-stream Siamese network which will be used to obtain the features descriptors and estimate scores for the corresponding images.

6.4 Experiments

6.4.1 Datasets and Experimental results

Experimental details.

All of our networks are implemented using the PyTorch [86] deep learning framework. For the backbone of our model, we use MobileNet [92], VGG-16 [98], and ResNet-50 [33]. The training

is performed on an NVIDIA Titan X GPU (12 GB of VRAM) paired with an Intel 8700k CPU with 32 GB of RAM memory. During training, before each image is fed into the network it goes through a set of augmentations consisting of random variations in image contrast, hue, saturation, and horizontal flipping. Then, it is stretched / compressed by resizing both the height and width to either 224, 256, 288 pixels. The networks are trained using stochastic gradient descent (SGD) with a starter learning rate of 0.0002 for 20 epochs, weight decay of 10^{-4}, and momentum of 0.9, The learning rate was halved every five epochs. The batch size is kept uniform across all studies to 32 (larger image sizes did not allow larger batches).

Evaluation.

Traditional datasets for image cropping (such as ICDB[117], and FCDB [15]) provided their test evaluation sets in the form of an image and a ground-truth bounding box annotation, which was supposed to represent the most pleasing crop as labeled by human annotators. Using the "best crop" bounding box and the predicting crop, previous works [16, 58, 110] compute the mean overlap and average boundary displacement

$$mIoU = \frac{1}{N}\sum_{i=1}^{N}\frac{area(\hat{C}_i \cap C_i)}{area(\hat{C}_i \cap C_i)} \tag{6.3}$$

and

$$mDisp. = \frac{1}{4N}\sum_{i=1}^{N}\sum_{j=1}^{4}\|\hat{B}_i^j - B_i^j\| \tag{6.4}$$

where $\hat{C}_i$ is is the estimated "best crop" and C_i is the ground truth crop respectively for image i, $\hat{B}_i^j - B_i^j$ represents the difference between the estimated and ground truth crops of a j^{th} boundary of the i^{th} image. As our main focus is in training composition ranking functions, for all of our experiments we compute and report the evaluation metrics of [119], which are more suited as ranking metrics. In the GAICD dataset, each image I_i comes with a set of proposals which have all been assigned a mean opinion score (MOS). These scores define the ranking between all of the proposals. Similarly to [119], let these ground truth scores be defined as $\mathbf{g}_i$. Let $\mathbf{p}_i$ be the set of predicted composition scores for the i^{th} image for the given proposals. Then the paper defines a metric

$$\overline{SRCC} = \frac{1}{N}\sum_{i=1}^{N} SRCC(\mathbf{g}_i, \mathbf{p}_i) \tag{6.5}$$

where $SRCC$ is the Spearman's rank order correlation between, in this case, the ground truth and predicted scores for image i. Furthermore, [119] defines a metric they call "return K of top-N

accuracy" ($Acc_{K/N}$), which is similar to the $Top-K$ accuracy / $Precision@K$ in image retrieval literature. It is defined as

$$Acc_{K/N} = \frac{1}{TK} \sum_{i=1}^{T} \sum_{j=1}^{K} True(c_{ij} \in S_i(N)), \tag{6.6}$$

where the function $True$ takes a value of 1 if its argument is $True$ else takes a value of 0, c_{ij} is the j^{th} crop of i^{th} image, and $S_i(N)$ is a set of crops which rank among top-N crops.

Dataset

Grid Anchor based Image Cropping Database (GAICD) consists of a total of 1,236 images (1000 images whose composition can be improved and 236 with ideal composition). Given the Grid Anchor based approach described in [119], a total of 106,680 candidate crops of the 1,236 images are evaluated by 19 annotators.

6.4.2 Ablative studies

We first perform a set of ablative studies, in which we try to understand the impact of various design choices on the performance of the network.

Backbone

To investigate the effect of backbone on the ranking performance, we fix the image size at 224 pixels per side, use Global Average Pooling to pool convolution features, and use a two-layer multi-layer perceptron (MLP with hidden layers set to 1024 and 512 pixels) to process the features and predict a score $\phi(I)$ for image I. As we can see in Table 6.1, the ResNet50 architecture achieves better ranking results in $\overline{SRCC}$, $\overline{Acc_5}$, and $\overline{Acc_{10}}$, which is consistent with the intuition and results of [28] as the ResNet50 CNN achieves a much smaller Top-5 classification error on the ImageNet dataset as compared to other models. Because of the strong performance of the ResNet50 backbone, it is chosen as the backbone for further ablation experiments.

Batch Normalization

The second set of experiments explores the effect of freezing the batch-normalization statistics and convolution weights. Somewhat surprisingly, if the batch-normalization statistics are allowed to change during training, the ranking performance of the model suffers significantly as we can see in Table 6.2. Performance recovers once the batch-normalization statistics are frozen to the original values trained on ImageNet. Just for completeness, we freeze the convolution layers and we see

Table 6.1: Comparison of the effect backbone architecture has on the ranking performance of the GAIC dataset.

Backbone	$\overline{SRCC}$	$Acc_{1/5}$	$A_{2/5}$	$A_{3/5}$	$A_{4/5}$	$\overline{Acc}_5$	$Acc_{1/10}$	$A_{2/10}$	$A_{3/10}$	$A_{4/10}$	$\overline{Acc}_{10}$
MobileNet V2	0.699	46.0	46.0	44.8	42.4	44.8	65.0	**64.0**	62.2	59.8	62.7
VGG16	0.671	48.0	**48.0**	43.2	40.7	45.0	63.5	**64.0**	60.3	56.9	61.2
ResNet50	**0.703**	**50.0**	44.0	**46.5**	**45.0**	**46.4**	**67.5**	62.7	**63.0**	**61.6**	**63.7**

Table 6.2: Effect of freezing the batch-normalization updates and convolution features on the ranking performance.

BatchNorm	$\overline{SRCC}$	$Acc_{1/5}$	$A_{2/5}$	$A_{3/5}$	$A_{4/5}$	$\overline{Acc}_5$	$Acc_{1/10}$	$A_{2/10}$	$A_{3/10}$	$A_{4/10}$	$\overline{Acc}_{10}$
No CHNG	0.158	8.5	8.7	8.7	9.6	8.9	13.5	14.0	15.2	15.6	14.6
Frozen BN	**0.703**	**50.0**	**44.0**	**46.5**	**45.0**	**46.4**	**67.5**	**62.7**	**63.0**	**61.6**	**63.7**
Top-layer	0.548	28.5	29.5	28.5	27.1	28.4	43.5	42.7	40.7	39.9	41.7

Table 6.3: Comparison of effect of image size and pooling type on the performance.

Image size	$\overline{SRCC}$	$Acc_{1/5}$	$A_{2/5}$	$A_{3/5}$	$A_{4/5}$	$\overline{Acc}_5$	$Acc_{1/10}$	$A_{2/10}$	$A_{3/10}$	$A_{4/10}$	$\overline{Acc}_{10}$
224 Avg. Pool	0.703	50.0	44.0	46.5	45.0	46.4	67.5	62.7	63.0	61.6	63.7
256 Avg. Pool	0.701	51.0	44.3	43.0	41.9	45.1	64.5	61.8	62.2	60.8	62.3
288 Avg. Pool	**0.727**	51.0	50.0	46.7	**45.4**	48.3	65.0	**66.0**	**63.8**	**62.1**	**64.2**
224 Flatten	0.705	52.0	49.2	45.8	44.1	47.8	**68.0**	63.7	61.5	61.1	63.6
256 Flatten	0.714	**52.5**	**50.7**	**49.7**	45.1	**49.5**	64.0	65.2	**63.8**	61.0	63.5
288 Flatten	0.696	50.0	46.5	45.5	41.5	45.9	65.0	64.0	62.7	59.0	62.7

Table 6.4: The effect of reducing the dimension of convolutional features on ranking performance.

Embedding dim.	$\overline{SRCC}$	$Acc_{1/5}$	$A_{2/5}$	$A_{3/5}$	$A_{4/5}$	$\overline{Acc}_5$	$Acc_{1/10}$	$A_{2/10}$	$A_{3/10}$	$A_{4/10}$	$\overline{Acc}_{10}$
256	**0.732**	52.0	48.2	47.7	45.6	48.4	69.0	66.5	65.3	63.9	66.2
128	0.715	49.5	48.5	47.0	45.1	47.5	66.0	65.7	64.3	62.0	64.5
32	0.713	**58.0**	52.7	48.3	46.0	51.2	69.0	67.2	63.8	62.7	65.7
8	0.715	56.5	**53.2**	**50.0**	**47.9**	**51.9**	**70.5**	**68.2**	**66.8**	**65.4**	**67.7**

Table 6.5: The effect of adding image blurring as a pre-processing step on the ranking performance.

Embedding dim.	$\overline{SRCC}$	$Acc_{1/5}$	$A_{2/5}$	$A_{3/5}$	$A_{4/5}$	$\overline{Acc}_5$	$Acc_{1/10}$	$A_{2/10}$	$A_{3/10}$	$A_{4/10}$	$\overline{Acc}_{10}$
256F	0.714	52.5	50.7	49.7	45.1	49.5	64.0	65.2	63.8	61.0	63.5
256FB3	**0.725**	51.5	50.0	**48.5**	**46.6**	49.1	67.0	68.0	65.8	64.5	66.3
256FB5	0.721	51.0	47.7	46.7	44.1	47.4	71.5	66.2	65.3	62.6	66.4
288A	0.727	51.0	50.0	46.7	45.4	48.3	65.0	66.0	63.8	62.1	64.2
288AB3	0.718	49.5	50.2	46.8	44.8	47.8	**73.0**	70.5	**68.2**	**66.0**	**69.4**
288AB5	**0.731**	**54.5**	**53.0**	48.3	45.1	**50.2**	72.0	**70.8**	67.7	65.5	69.0

that we do not achieve as strong of a performance as the convolution weights are not allowed to change.

Image size and pooling type

Further, we compare the effect of input image size and pooling type on the ranking of performance. This is motivated by the observation that in composition, all parts of the image play a role in its composition quality. Therefore we consider the image sizes of 224, 256, and 288 pixels on the side. Then, we either use Global Average Pooling (Avg) to pool features across the spatial dimension and flatten the convolution feature map and feed it into the MLP. Though there is no clear winner among the combinations of the image size / pooling combinations - 288 / Avg and 256 / Flatten - achieve the best performance given the pooling type.

Convolution features dimension reduction

Opting to flatten the convolution features to consider all spatial locations incurs significant computational penalty. To combat that, we evaluate the possibility of adding a fully connected layer that reduces the dimension f of the final convolution layer. This operation is implemented as a $1 \times 1 \times c$ convolution, where c is the dimension of the final layer, reducing the dimension from $H \times C \times f$ to $H \times C \times c$. Overall, we can observe that dimension reduction does not improve $\overline{SRCC}$, however $\overline{Acc}_{10}$ is boosted. The boost in $\overline{Acc}_{10}$ can be attributed to the additional trainable parameters in the convolution layer.

Blurring the images

Composition is not as dependent on the notion of sharpness as aesthetics is and therefore we investigate adding Gaussian blurring as a pre-processing step before feeding images into the network. As we can see, in both scenarios - 288 / Avg and 256 / Flatten - the $\overline{Acc}_{10}$ is significantly boosted.

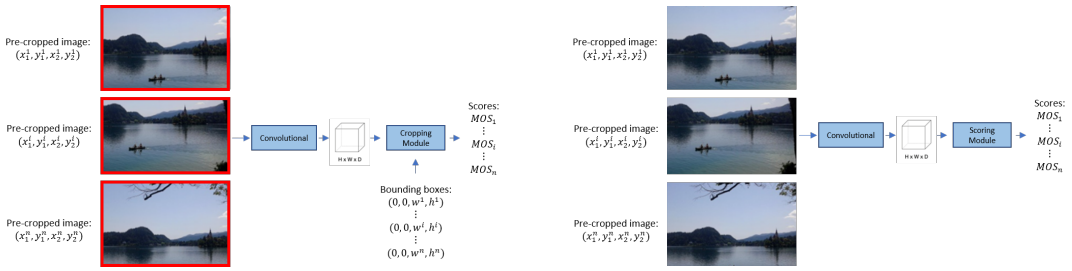
6.4.3 Comparison to the state-of-the-art

Comparison models

This section describes the models we and [119] compare against.

A2-RL [61] model uses reinforcement learning (RL) to train an agent that, when applied to an image, chooses one of 16 different actions (e.g. shifting and scaling) to propose a sequence of actions to crop the image.

View Finding Network (VFN) [16] is a model based on the AlexNet architecture, and uses professional photographs and its sub-crops to train a pair-wise ranking network to enforce an underlying assumption: a sub-crop of a professional image is of worse composition quality.



(a) Evaluation of GAIC model.

(b) Evaluation of RankNet model.

Figure 6.4: Figure showing high-level description our modified GAIC model vs our approach.

Method	$\overline{SRCC}$	$Acc_{1/5}$	$A_{2/5}$	$A_{3/5}$	$A_{4/5}$	$\overline{Acc_5}$	$Acc_{1/10}$	$A_{2/10}$	$A_{3/10}$	$A_{4/10}$	$\overline{Acc_{10}}$
A2-RL [61]	-	23.0	-	-	-	-	-	-	-	-	-
VPN [110]	-	40.0	-	-	-	-	-	-	-	-	-
VFN [16]	0.450	27.0	28.0	27.2	24.6	26.7	39.0	39.3	39.0	37.3	38.7
VEN [119]	0.621	40.5	36.5	36.7	36.8	37.6	54.0	51.0	50.4	48.4	50.9
GAIC-VGG-Or-v1 [119]	0.729	49.5	48.7	45.8	44.0	47.0	67.5	65.7	63.2	61.9	64.6
GAIC-RN50-Or-v1 [119]	0.725	50.0	50.2	**49.5**	45.8	48.9	70.5	**71.3**	**68.2**	65.2	68.8
GAIC-VGG-S-v1 [119]	0.738	55.0	49.7	46.3	45.4	49.1	70.0	68.2	66.2	65.4	67.4
GAIC-RN50-S-v1 [119]	0.689	45.5	43.5	43.2	42.4	43.6	62.0	63.0	63.0	62.0	62.5
GAiC-VGG-Or-v2 [119]	0.342	20.5	18.8	20.7	20.3	20.1	28.5	30.0	30.5	31.0	30.0
GAIC-RN50-Or-v2 [119]	0.382	22.5	23.3	23.0	21.0	22.5	31.0	33.3	34.8	32.1	32.8
GAIC-VGG-S-v2 [119]	0.227	18.5	16.5	16.7	17.2	17.2	26.0	26.5	27.3	27.9	26.9
GAIC-RN50-S-v2 [119]	0.298	22.0	21.7	21.2	20.3	21.3	33.5	33.3	32.2	31.5	32.6
RankNet256F	0.729	**58.0**	**53.0**	48.7	**47.2**	**51.7**	**73.5**	69.3	67.0	**65.5**	68.8
RankNet288A	0.731	54.5	**53.0**	48.3	45.1	50.2	72.0	70.8	67.7	**65.5**	**69.0**

Table 6.6: Quantitative comparison of our best RankNet models to state-of-the-art models on the GAICD dataset. The GAIC model is described as GAIC-backbone-features-evaluation, where the backbone is set to either VGG16 or Resnet50, and features considered for predicting composition score are that of both foreground / background (Or) or just the foreground (S). Evaluation types denoted by v1 and v2 correspond to the original evaluation of [119] and modified respectively. The region delineated by the bounding box is considered as the foreground. For further description of individual models, and modified evaluation paradigm, please see Section 6.4.3.

View Evaluation Network (VEN) [110] is a model based on the VGG16 architecture, and uses the CPC dataset introduced by Wei et al. [110] to train a pair-wise ranking network.

View Proposal Network (VPN) [110] is a model based on a Single Shot Detector detection model, and is used in tandem with the VEN, trained via a Student-Teacher framework, where VEN is used as a weak supervisor to encourage proposed views with good composition scores as judged by VEN.

Grid-anchor based Image Cropping (GAIC) [119] is an image cropping model inspired by image detection models. Given an image, its convolutional feature representation and a bounding box are fed into a cropping module. The module uses the RoI align operation introduced by He et al. [31] to construct a representation which considers the features from the bounding box as well as the discarded region outside of the box. It is very efficient as it only extracts the convolutional feature map once, and then predicts scores for any number of bounding boxes (regular grid in case of GAIC). This model is denoted as *GAIC-backbone-Or*. In addition to these models, we also consider a set of models denoted as *GAIC-backbone-S*, in which only the features from the bounding box are used to predict the composition scores. Furthermore, each of the models are evaluated in two different regimes: one described by [119], and one which resembles our evaluation framework - to obtain a score for a given bounding box, the image is pre-cropped and fed into the GAIC model, where the bounding box is set to the width and height of the box. Please see Figure 6.4 for further illustration of parallels between GAIC evaluated using the second evaluation method and our model.

Quantitative results

The GAIC [119] PyTorch[1] implementation does not specify a validation set and thus it is not clear how the best models is chosen. Thus for fairness in comparison, we used the implementation to retrain the models using our training set, and choose the best model according to the hold-out / validation set. These models are then used to obtain the test scores in Table 6.6.

Table 6.6 compares our best RankNet models to the current and previous state-of-the-art models. As we can see in Table 6.6, both the ResNet50 (GAIC-RN50-Or) and VGG16 (GAIC-VGG-Or) models achieve very comparable or better results as compared to the ones reported in [119]. With the goal of training functions which aim to provide us with both composition scores or image features, we consider an alternative way of evaluating the GAIC model tailored to better resemble our model (described above).

As we can see from the results in Table 6.6, our approach allows us to train state-of-the-art ranking networks as compared to the previous state of the art model of [119] denoted by GAIC-VGG-Or-v1. In this case, our RankNet models match GAIC-VGG-Or-v1 in $\overline{SRCC}$, and outperforms it in terms of $\overline{Acc}_5$ and $\overline{Acc}_{10}$.

[1] https://github.com/HuiZeng/Grid-Anchor-based-Image-Cropping-Pytorch

(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 6.5: Figure showing qualitative result for RankNet. Figure on the left shows the image with the best ground truth ranked sub-crop, and the rest of images show the image overlaid with the bounding box that was ranked to be : best, 66th percentile, 33th percentile, worst from left to right by our model.

Interestingly, some of the results are changed if the backbone or features set used to predict the scores is changed. First, we notice that swapping the backbone for ResNet50 negligibly deteriorates the $\overline{SRCC}$, however significantly improves the $\overline{Acc_5}$ and $\overline{Acc_{10}}$. Additionally, contrary to intuition, when using the VGG backbone, only considering the foreground features improves all of the metrics, though this is not the case for ResNet50. Lastly, we can see from the results, once we transform the evaluation of GAIC model to better resemble our problem, the performance of the GAIC approach suffers considerably.

6.5 Conclusion

In our paper, we explore the key choices critical for learning state-of-the-art composition ranking functions. We further show that though the cropping model of [119] does well in ranking the crops, its performance severely deteriorates. Contrary to the claims of [119], we show that through careful architecture choices, data sampling methods and data augmentations, one can train a state-of-the-art ranking CNN, which has potential uses in image cropping, as a teacher model for providing soft supervision, and for extracting composition features for the image.

Chapter 7

Conclusion and Future Work

This dissertation explores representation learning and various representations for predicting image aesthetics and composition quality of the images, which are eventually used for image cropping.

In **Chapter 4** (and partially **Chapter 3**) we have explored expert (i.e. hand-crafted) features and their ability to inform generic and fine-tuned deep learning (DL) features, with the goal of bridging the gap between early work using hand-crafted features and deep learning. Initially, we showed that (a) a combination of HC features can compete with DL features, and (b) many of the hand-crafted features capture complimentary information (despite removing a subset of the features, the predictive performance of the remaining features did not deteriorate). We showed DL features can be augmented via fusion (early and late) with HC features. Features that measure technical quality, photographic-rules (e.g. rule of thirds), and color information were most important for improving fusion models.

In **Chapter 5**, we discussed the performance of deep learning representations learned by constrained models, motivated by the desire to perform aesthetic inference and ranking on mobile devices. We showed that rank-order correlation of a model is proportional to its complexity (number of operations the model executes computing a score), which is controlled through the choice of input size and model depth. We saw, that an aesthetic ranking function can achieve near state-of-the-art results in aesthetic cropping, however it lags behind composition ranking functions.

With the recent release of appropriate datasets and motivated by the lack of exploration in principled composition ranking, with its potential use for image cropping, in **Chapter 6** we studied various architecture and training choices to identify key steps that allow us to train state-of-the-art composition ranking functions. We showed pair-wise ranking optimization allows us to train ranking functions, which are more robust as compared to some of the previous cropping models.

In this dissertation we explored ways of improving and learning representations for aesthetic inference and image enhancement. Through our exploration we have uncovered possible directions that can build upon the work in this dissertation and we outline them in the following section.

7.1 Future Work

7.1.1 Modeling individual aesthetic preferences

The majority of work up to present focused on modeling course aesthetic preferences, primarily via summarizing the distribution of user ratings by simpler statistics such as the mean. Then various models were created to either predict the mean score or predict aesthetic classes (obtained by binning the images into coarser classes based on the said mean). Capturing the preferences of a single user is often a challenging problem because of: (a) the scarcity of labeled data that would directly tell us what the user prefers, (b) difficulty of learning from small datasets, or (c) concerns related to privacy when collecting data from an individual. However, many of these concerns could find solutions in the near future. For example, recent lifelong learning approaches can help us with learning from small datasets, while federated learning can help us with issues relating to privacy. Being able to predict personal preferences of users can help us created tailored experiences for users, or as a way to predict soft biometrics as shown by Segalin et al. [95]. It would be interesting to explore temporal modeling of user preferences and if such understanding could help us predict depression in users.

7.1.2 Improving representation learning for composition ranking

In **Chapter 6** we briefly discussed the challenges of ranking (learning ranking functions with pairwise loss functions, or representation learning using triplet-losses) and focused on understanding key choices in network architectures for learning composition ranking functions. In the recent years many improvements in representation learning for image retrieval have relied on devising better methods to obtain labels / correspondences [90] and improving n-tuplet sampling [116]. For composition ranking, one possible approach to obtain better labels and data pairs would utilize the ideas from curriculum learning [7] and pseudo-labels [60] to help us guide the learning process and automatically gather more training data labels.

7.1.3 Empirical understanding of image cropping

In our previous work, and from the literature [119], we saw that models trained on various cropping datasets do not generalize well to the other datasets. Similar problems have been shown in other problem domains such as Visual Question Answering [96], where Shreshta et al. show the lack of generalization between datasets that focus on synthetic versus natural images. Therefore to further understanding of image cropping and the reason for the gap in generalization across datasets, we deem it necessary to better explore the differences in the sources of data in the sets, challenges with labeling, establish which datasets provide models that appeal to users better, and explore approaches allowing datasets to inform each other.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, 2009.

[3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[4] S. Alpert, M. Galun, A. Brandt, and R. Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):315–327, 2012.

[5] T. Ang. *Digital Photographer's Handbook*. DK ADULT, 5th edition, 2012.

[6] T. Aydin, A. Smolic, and M. Gross. Automated aesthetic analysis of photographic images. *Visualization and Computer Graphics, IEEE Transactions on*, 21(1):31–42, Jan 2015.

[7] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA, 2009. ACM.

[8] S. Bhattacharya, B. Nojavanasghari, T. Chen, D. Liu, S.-F. Chang, and M. Shah. Towards a comprehensive computational model for aesthetic assessment of videos. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 361–364, New York, NY, USA, 2013. ACM.

[9] S. Bhattacharya, R. Sukthankar, and M. Shah. A framework for photo-quality assessment and enhancement based on visual aesthetics. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 271–280, New York, NY, USA, 2010. ACM.

[10] A. Borji, D. N. Sihite, and L. Itti. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, 22(1):55–69, 2013.

[11] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, page 737–744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[12] C. D. Cerosaletti and A. C. Loui. Measuring the perceived aesthetic quality of photographic images. In *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*, pages 47–52, July 2009.

[13] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai. Fusing generic objectness and visual saliency for salient object detection. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, page 914–921, USA, 2011. IEEE Computer Society.

[14] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[15] Y. Chen, T. Huang, K. Chang, Y. Tsai, H. Chen, and B. Chen. Quantitative analysis of automatic image cropping algorithms: A dataset and comparative study. *CoRR*, abs/1701.01480, 2017.

[16] Y. Chen, J. Klopp, M. Sun, S. Chien, and K. Ma. Learning to compose with professional photographs on the web. *CoRR*, abs/1702.00503, 2017.

[17] B. Cheng, B. Ni, S. Yan, and Q. Tian. Learning to photograph. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 291–300, New York, NY, USA, 2010. ACM.

[18] M. Cheng, G. Zhang, N. J. Mitra, X. Huang, and S. Hu. Global contrast based salient region detection. In *CVPR 2011*, pages 409–416, 2011.

[19] B. C. Csáji et al. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24(48):7, 2001.

[20] R. Datta, D. Joshi, J. Li, and J. Z. Wang. *Studying Aesthetics in Photographic Images Using a Computational Approach*, pages 288–301. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[21] R. Datta, J. Li, and J. Z. Wang. Learning the consensus on visual quality for next-generation image management. In *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, pages 533–536, New York, NY, USA, 2007. ACM.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[23] Y. Deng, C. C. Loy, and X. Tang. Image aesthetic assessment: An experimental survey. *CoRR*, abs/1610.00838, 2016.

[24] S. Dhar, V. Ordonez, and T. Berg. High level describable attributes for predicting aesthetics and interestingness. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1657–1664, June 2011.

[25] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

[26] DMLC. *Understand your dataset with XGBoost*, 2016 (accessed September 20, 2016).

[27] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2376–2383, 2010.

[28] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *CoRR*, abs/1610.07940, 2016.

[29] L. Guo and F. Li. Image aesthetic evaluation using paralleled deep convolution neural network. *CoRR*, abs/1505.05225, 2015.

[30] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, Jan 2002.

[31] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[32] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2341–2353, Dec. 2011.

[33] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[34] S. Heyman. Photos, photos everywhere, Jul 2015.

[35] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. *ArXiv e-prints*, Mar. 2015.

[36] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[37] Z. Y. T. L. Huaizu Jiang, Jingdong Wang and N. Zheng. Automatic salient object segmentation based on context and shape prior. In *Proc. BMVC*, pages 110.1–110.12, 2011. http://dx.doi.org/10.5244/C.25.110.

[38] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[39] M. Jaderberg, A. Vedaldi, and A. Zisserman. Speeding up convolutional neural networks with low rank expansions. *CoRR*, abs/1405.3866, 2014.

[40] Z. Jenkins. Top devices of 2017 on flickr — flickr blog, Dec 2017.

[41] W. Jiang, A. C. Loui, and C. D. Cerosaletti. Automatic aesthetic value assessment in photographic images. In *2010 IEEE International Conference on Multimedia and Expo*, pages 920–925, July 2010.

[42] J. Jin, A. Dundar, and E. Culurciello. Flattened convolutional neural networks for feedforward acceleration. *CoRR*, abs/1412.5474, 2015.

[43] X. Jin, J. Chi, S. Peng, Y. Tian, and C. Y. andXiaodong Li. Deep image aesthetics classification using inception modules and fine-tuning connected layer. In *8th International Conference on Wireless Communications & Signal Processing, WCSP 2016, Yangzhou, China, October 13-15, 2016*, pages 1–6, 2016.

[44] D. Joshi, R. Datta, E. Fedorovskaya, Q. T. Luong, J. Z. Wang, J. Li, and J. Luo. Aesthetics and emotions in images. *IEEE Signal Processing Magazine*, 28(5):94–115, Sept 2011.

[45] Y. Kao, R. He, and K. Huang. Visual aesthetic quality assessment with multi-task deep learning. *CoRR*, abs/1604.04970, 2016.

[46] Y. Kao, K. Huang, and S. Maybank. Hierarchical aesthetic quality assessment using deep convolutional neural networks. *Image Commun.*, 47(C):500–510, Sept. 2016.

[47] Y. Ke, X. Tang, and F. Jing. The design of high-level features for photo quality assessment. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 419–426, June 2006.

[48] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[49] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks, 2016.

[50] C. Koch and S. Ullman. *Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry*, pages 115–141. Springer Netherlands, Dordrecht, 1987.

[51] R. Kohavi. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. PhD thesis, Stanford, CA, USA, 1996. UMI Order No. GAX96-11989.

[52] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273 – 324, 1997. Relevance.

[53] S. Kong, X. Shen, Z. Lin, R. Mech, and C. C. Fowlkes. Photo aesthetics ranking network with attributes and content adaptation. *CoRR*, abs/1606.01621, 2016.

[54] S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? 2019.

[55] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[56] M. Kucer, N. D. Cahill, A. C. Loui, and D. W. Messinger. Augmenting salient foreground detection using fiedler vector for multi-object segmentation. *Electronic Imaging*, 2017(17):116–121, 2017.

[57] M. Kucer, A. C. Loui, and D. W. Messinger. Leveraging expert feature knowledge for predicting image aesthetics. *IEEE Transactions on Image Processing*, 27(10):5100–5112, Oct 2018.

[58] M. Kucer and D. W. Messinger. Aesthetic inference for smart mobile devices. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1764–1773, March 2018.

[59] P. Labs. iphone x benchmarks - geekbench browser, 2017. [Online; accessed 1-December-2017].

[60] D.-H. Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. 2013.

[61] D. Li, H. Wu, J. Zhang, and K. Huang. A2-rl: Aesthetics aware reinforcement learning for automatic image cropping. *arXiv preprint arXiv:1709.04595*, 2017.

[62] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. *CoRR*, abs/1608.08710, 2016.

[63] J. Li, M. D. Levine, X. An, X. Xu, and H. He. Visual saliency based on scale-space analysis in the frequency domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):996–1010, 2013.

[64] D. Liu, R. Puri, N. Kamath, and S. Bhattacharya. Composition-aware image aesthetics assessment. *CoRR*, abs/1907.10801, 2019.

[65] F. Liu and S. Osindero. A complete framework for aesthetic inference in images. *arXiv*, 2015.

[66] A. Loui, M. D. Wood, A. Scalise, and J. Birkelund. Multidimensional image value assessment and rating for automated albuming and retrieval. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 97–100, Oct 2008.

[67] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts. Understanding variable importances in forests of randomized trees. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 431–439. Curran Associates, Inc., 2013.

[68] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang. Rapid: Rating pictorial aesthetics using deep learning. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 457–466, New York, NY, USA, 2014. ACM.

[69] X. Lu, Z. Lin, X. Shen, R. Mech, and J. Z. Wang. Deep multi-patch aggregation network for image style, aesthetics, and quality estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 990–998, Dec 2015.

[70] F. P. X. X. G. Luca Marchesotti (Xerox). Learning beautiful (and ugly) attributes. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013.

[71] W. Luo, X. Wang, and X. Tang. Content-based photo quality assessment. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2206–2213, Nov 2011.

[72] Y. Luo and X. Tang. Photo and video quality evaluation: Focusing on the subject. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, pages 386–399, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[73] S. Ma, J. Liu, and C. W. Chen. A-lamp: Adaptive layout-aware multi-patch deep convolutional neural network for photo aesthetic assessment. *CoRR*, abs/1704.00248, 2017.

[74] J. Machajdik and A. Hanbury. Affective image classification using features inspired by psychology and art theory. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 83–92, New York, NY, USA, 2010. ACM.

[75] L. Mai, H. Jin, and F. Liu. Composition-preserving deep photo aesthetics assessment. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 497–506, June 2016.

[76] L. Marchesotti, N. Murray, and F. Perronnin. Discovering beautiful attributes for aesthetic image analysis. *Int. J. Comput. Vision*, 113(3):246–266, July 2015.

[77] L. Marchesotti, F. Perronnin, D. Larlus, and G. Csurka. Assessing the aesthetic quality of photographs using generic image descriptors. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 1784–1791, Washington, DC, USA, 2011. IEEE Computer Society.

[78] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. Pruning convolutional neural networks for resource efficient transfer learning. *CoRR*, abs/1611.06440, 2016.

[79] J. Morton. Basic color theory, 2009. Available at `https://www.colormatters.com/color-and-design/basic-color-theory`.

[80] N. Murray, L. Marchesotti, and F. Perronnin. Ava: A large-scale database for aesthetic visual analysis. pages 2408–2415, June 2012.

[81] N. Murray, L. Marchesotti, and F. Perronnin. Learning to rank images using semantic and aesthetic labels. In *British machine and vision conference (BMVC)*, 2012.

[82] M. Nishiyama, T. Okabe, I. Sato, and Y. Sato. Aesthetic quality classification of photographs based on color harmony. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 33–40, June 2011.

[83] M. Nishiyama, T. Okabe, Y. Sato, and I. Sato. Sensation-based photo cropping. In *Proceedings of the 17th ACM International Conference on Multimedia*, MM '09, pages 669–672, New York, NY, USA, 2009. ACM.

[84] P. Obrador, M. A. Saad, P. Suryanarayan, and N. Oliver. *Advances in Multimedia Modeling: 18th International Conference, MMM 2012, Klagenfurt, Austria, January 4-6, 2012. Proceedings*, chapter Towards Category-Based Aesthetic Models of Photographs, pages 63–76. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[85] P. Obrador, L. Schmidt-Hackenberg, and N. Oliver. The role of image composition in image aesthetics. In *2010 IEEE International Conference on Image Processing*, pages 3185–3188, Sep. 2010.

[86] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[87] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 733–740, 2012.

[88] F. Perazzi, O. Sorkine-Hornung, and A. Sorkine-Hornung. Efficient salient foreground detection for images and video using fiedler vectors. In *Proceedings of the Eurographics Workshop on Intelligent Cinematography and Editing*, WICED '15, page 21–29, Goslar, DEU, 2015. Eurographics Association.

[89] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[90] F. Radenović, G. Tolias, and O. Chum. Fine-tuning CNN image retrieval with no human annotation. *TPAMI*, 2018.

[91] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, Nov. 1987.

[92] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018.

[93] A. E. Savakis, S. P. Etz, and A. C. P. Loui. Evaluation of image appeal in consumer photography, 2000.

[94] R. Schifanella, M. Redi, and L. M. Aiello. An image is worth more than a thousand favorites: Surfacing the hidden beauty of flickr pictures. In *ICWSM'15: Proceedings of the 9th AAAI International Conference on Weblogs and Social Media*. AAAI, 2015.

[95] C. Segalin, A. Perina, and M. Cristani. Personal aesthetics for soft biometrics: A generative multi-resolution approach. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, page 180–187, New York, NY, USA, 2014. Association for Computing Machinery.

[96] R. Shrestha, K. Kafle, and C. Kanan. Answer them all! toward universal visual question answering models. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10464–10473, 2019.

[97] F. Simond, N. Arvanitopoulos, and S. Süsstrunk. Image aesthetics depends on context. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3788–3792, Sept 2015.

[98] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[99] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[100] X. Tang, W. Luo, and X. Wang. Content-based photo quality assessment. *Multimedia, IEEE Transactions on*, 15(8):1930–1943, Dec 2013.

[101] L. Theis, I. Korshunova, A. Tejani, and F. Huszár. Faster gaze prediction with dense networks and fisher pruning. *CoRR*, abs/1801.05787, 2018.

[102] L. Theis and Z. Wang. Speedy neural networks for smart auto-cropping of images. Accessed: 2019-06-01.

[103] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li. The new data and new challenges in multimedia research. *CoRR*, abs/1503.01817, 2015.

[104] H. Tong, M. Li, H. Zhang, and C. Zhang. Blur detection for digital images using wavelet transform. In *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, volume 1, pages 17–20 Vol.1, June 2004.

[105] V. Vanhoucke, A. Senior, and M. Z. Mao. Improving the speed of neural networks on cpus. In *Deep Learning and Unsupervised Feature Learning Workshop, NIPS 2011*, 2011.

[106] W. Wang, M. Zhao, L. Wang, J. Huang, C. Cai, and X. Xu. A multi-scene deep learning model for image aesthetic evaluation. *Image Commun.*, 47(C):511–518, Sept. 2016.

[107] Y. Wang, Z. Lin, X. Shen, R. Mech, G. S. P. Miller, and G. W. Cottrell. Recognizing and curating photo albums via event-specific image importance. *CoRR*, abs/1707.05911, 2017.

[108] Z. Wang, F. Dolcos, D. Beck, S. Chang, and T. S. Huang. Brain-inspired deep networks for image aesthetics assessment. *ArXiv e-prints*, Jan. 2016.

[109] Y. Wei, F. Wen, W. Zhu, and J. Sun. Geodesic saliency using background priors. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, page 29–42, Berlin, Heidelberg, 2012. Springer-Verlag.

[110] Z. Wei, J. Zhang, X. Shen, Z. Lin, R. Mech, M. Hoai, and D. Samaras. Good view hunting: Learning photo composition from dense view pairs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[111] Wikipedia. Iphone x — wikipedia, the free encyclopedia, 2017. [Online; accessed 1-December-2017].

[112] Wikipedia. Nokia 7650 — wikipedia, the free encyclopedia, 2017. [Online; accessed 1-December-2017].

[113] Wikipedia contributors. Softmax function — Wikipedia, the free encyclopedia, 2020. [Online; accessed 24-June-2020].

[114] Wikipedia contributors. Stochastic gradient descent — Wikipedia, the free encyclopedia, 2020. [Online; accessed 24-June-2020].

[115] L.-K. Wong and K.-L. Low. Saliency-enhanced image aesthetics class prediction. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 997–1000, Nov 2009.

[116] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krähenbühl. Sampling matters in deep embedding learning. In *ICCV*, 2017.

[117] J. Yan, S. Lin, S. B. Kang, and X. Tang. Learning the change for automatic image cropping. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 971–978, June 2013.

[118] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang. Saliency detection via graph-based manifold ranking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3166–3173, 2013.

[119] L. L. C. Z. Zeng, Hui and L. Zhang. Reliable and efficient image cropping: A grid anchor based approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[120] L. Zhang, Y. Gao, R. Zimmermann, Q. Tian, and X. Li. Fusion of multichannel local and global structural cues for photo aesthetics evaluation. *IEEE Transactions on Image Processing*, 23(3):1419–1429, March 2014.

[121] J. Zuckerman. Jim zuckerman on composition: Symmetry, 2017.

[122] J. Zuckerman. Jim zuckerman on composition: The rule of thirds, 2017.

[123] D. Zwillinger and S. Kokoska. Crc standard probability and statistics tables and formulae. 01 2000.